# Linked Data in VRA Core 4.0: Converting VRA XML Records into RDF/XML

A thesis submitted to the College of Communication and Information of Kent State University in partial fulfillment of the requirements for the Master of Library and Information Science and Master of Science dual degree program

By

Jeffrey Mixter

May, 2013

Thesis written by

Jeffrey Mixter

B.A., Ohio State University, 2010

M.S., Kent State University, 2013

M.L.I.S., Kent State University, 2013

Approved by

______________________________________

Marcia Lei Zeng, Ph.D., Advisor

______________________________________

Tomas A. Lipinski, J.D., LL.M., Ph.D., Director, School of Library and Information Science

______________________________________

Stanley T. Wearden, Ph.D., Dean, College of Communication and Information

## Table of Contents

Page

## List of Figures

Figure Page

## List of Tables

# Acknowledgments

I would like to thank all of those who have helped and encouraged me throughout my work on this study. Specifically my wonderful wife Michelle, who put up with so much, Dr. Marcia Zeng, my thesis advisor, and the other members of the thesis advisory board, Dr. Athena Salaba and Dr. Yin Zhang.

Special thanks go to Jeff Young, an OCLC co-worker and friend, who helped review my results and provided invaluable insight and opinions on Linked Data

# Chapter I

## Introduction

Over past decades, libraries have built and collected an immense amount of data. Although this information is very important and has been well maintained, it traditionally has been stuck in the information silos of the individual organizations. The development of online union catalogs introduced the practice of copy cataloging and record sharing, which allowed libraries to share bibliographic information rather than creating their own independent records. Similarly, recent developments in the Semantic Web have made it possible to share entire datasets of valuable information over the Internet, and consequently with the entire world. Organizations such as OCLC Online Computing Center, The New York Times, the BBC, the Library of Congress and the Smithsonian Institution all have begun to share their information as Linked Data. Even more importantly, the sharing of this information has allowed organizations such as Freebase, DBpeida.org and OCLC to begin to use the Linked Open Data information to develop tools that aggregate and share, from their hub locations, all of the valuable information that they previously had stored in internal database silos. It should be stated that although there has been immense interest in publishing Linked Open Data, there have been few (if any) notable empirical studies examining the value of doing so. In addition to the Semantic Web allowing organizations to share valuable data online, the use of uniform/standard methods has allowed that data to be interconnected and used with other datasets.

### Background

There have been recent attempts by museums to incorporate RDF triples into visual image cataloging data models. The International Council of Museums has published the CIDOC Conceptual Reference Model (CRM), and Europeana (the European Digital Library) has published its own RDF vocabulary. In the attempt to describe as accurately as possible not only images, but also cultural items, sound recordings, and videos, both the CIDOC model and the Europeana model have proven to be very complex and intricate (Patel et al., 2005). The relative complexity of the CIDOC model has made it difficult to use and incorporate existing standards. Additionally, neither the CIDOC model nor Europeana's model attempt to use any existing formats. Instead, they have developed an entirely unique vocabulary for publishing their information as Linked Open Data. Although their data models do allow information to be published in the Semantic Web, the lack of mappings to standard vocabularies causes problems with regards to linking to, and being linked to by, similar datasets.

The Visual Resources Association Core Categories (VRA Core) 4 XML schema was developed by the Visual Resources Association. Since its endorsement by the METS Editorial Board in 2007, 56 organizations have adopted VRA 4.0. Most notably, Cornell University, North Carolina State University, Rhode Island School of Design, Smith College, the University of Cincinnati, the University of Michigan and the University of York all have adopted VRA 4.0 and made their various collections available through the VRA 4.0 Implementation Registry website. It is not clear, though, if the XML elements and attributes actually are being used in their records, or whether the XML simply is being used as a file format for data transfer by these VRA adopters. The University of Cincinnati and Rhode Island School of Design both use specific elements of the VRA Core 4.0. The University of Michigan, the University of York and North Carolina State University all use the full VRA 4.0 schema. Smith College uses VRA 4.0 as a schema, but ignores some of the more granular attributes. Cornell University uses the VRA XML schema primarily as a tool for mapping records over to LUNA, an image management system, although some newer collections are beginning to use VRA 4.0 as a cataloging data model (VRA, 2011).

## Problem Statement

Libraries, museums, and archives store a massive amount of valuable data in internal databases. These databases act as information silos, which prevents the information from being shared and consequently used by outside people or organizations. Developments in the Semantic Web now allow organizations to share and relate their data with and to the larger community. This community includes not only other libraries, museums and archives, but also other information organizations that have similar and related data. The World Wide Web Consortium (W3C) established a set of standards for sharing information on the Semantic Web. Data needs to be published as RDF triples in order for it to be consumed, understood, and interconnected with other Linked Open Data. Existing image cataloging modules are not designed to adhere to the Semantic Web standards set forth by the W3C. Similar to library MARC21 bibliographic records, images traditionally have been cataloged in formats that are not of use to organizations outside of the realm of library science. VRA Core 4.0 tried to address this issue by developing a XML schema. While this format allows VRA records to be shared, in a readable format, with the outside world, the simple XML still did not incorporate the rich relational elements and identifiers that could be used easily to connect VRA data with other related data.

The problem addressed in this study was how to develop and incorporate an automatic method for converting VRA Core 4.0 XML records into RDF. This study used VRA Core 4.0 (http://www.vraweb.org/index.html) as the basis for a working image cataloging data model. Additionally, the study used current popular vocabularies, primarily Schema.org (http://schema.org/), to conceptualize the VRA Core data model and to convert XML elements and attributes into RDF classes and properties. VRA Core was selected because it is a data standard for the description of works of visual culture, as well as for the images that document them. Another reason VRA Core was used as a data model is because it has not yet been transcribed into RDF. CIDOC already uses RDF, but the current model uses custom RDF elements, which are not mapped to other RDF vocabularies. The practice of not mapping custom vocabularies to other similar vocabularies causes problems with interoperability and hinders data sharing within the greater Semantic Web. Developing and using a model that incorporates popular vocabularies hopefully will allow the data to be shared more openly and allow for maximized interoperability. This is what OCLC has attempted to do in its implementation of Linked Data into WorldCat (OCLC, 2012a).

## Objectives

The primary objective of this study was to create an effective and efficient way to map the existing VRA 4.0 data model to a new proposed RDF data model. In so doing, it was important to take into account that the RDF must be rich enough to cover the functional requirements of museums, libraries and archives, yet also intuitive enough for novice catalogers to decipher and use. It also was important for the purposes of maximizing interoperability to use popular RDF vocabularies to the greatest extent possible.

In addition to developing a RDF data model, another strategic objective was to create a XSLT stylesheet that enables current users of VRA Core 4.0 automatically to convert records into RDF/XML (W3C, 2012b). This will serve as a way for organizations that currently are using VRA Core 4.0, but that might not have a solid understanding of RDF, easily to convert their legacy records into RDF/XML. Once the records are converted they will be able to be published and shared as Linked Data.

## Significance

This study helped develop a comprehensive and systematic way for libraries, museums and archives to convert all of their existing VRA 4.0 records into RDF/XML. The immediate benefit that provides is to allow catalogers to maintain existing cataloging practices while simultaneously being able to publish Linked Open Data to the Semantic Web. After the data is extracted from the existing information silos and published as Linked Open Data, it will be possible to create connections, both to and from the data, with other relevant datasets. The use of popular vocabularies helped in this effort by providing existing elements and URIs that can be used automatically to link across data silos.

Once the data is available as Linked Data, further development and studies can be conducted to create unique and valuable ways for using the data. Publishing data as Linked Data can lead both to internal as well as to external benefits for organizations. The major internal benefit of publishing Linked Data is that is can help eliminate problems associated with data silos. A problem that organizations frequently face is that data in one divisional database cannot be used by another division because the schemas do not match or the data model used is not compatible. Using Linked Data could allow for data to be standardized across the organization, thereby eliminating the time-consuming process of converting existing data into a useable format/model.

The immediate external benefit is that organizations will be able to link their data with other similar or related datasets. This can help improve visibility and use of data that otherwise would be stuck in data silos. Sharing data also can help improve the overall quality of data as well as make it more useful to end-users. The International System for Agricultural Science and Technology (AGRIS) has recently launched, as a beta release, a Linked Data-powered Web application called OpenAgris. It aggregates information from different Web sources to expand the AGRIS (http://aims.fao.org/openagris). It is able to generate a Linked Data mash-up that provides its users with additional information pertinent to the articles that they are reviewing. The beta version of OpenAgris uses datasets from DBPedia.org, the Global Biodiversity Information Facility (BGIF), FAO Geopolitical Ontology, AGRIS serials dataset, Global Capture Production and the World Bank to generate a more complete and information-rich experience for end-users.

## Definitions

The definitions provided in this section are to help familiarize readers with the specific terminology used in this thesis. In addition, since many of the terms below can be broadly defined and used in a wide variety of cases, these definitions also add specific context as to how each is used throughout the study.

**Data** – In this thesis "data" is used to refer to metadata created by library, museums and archives.

**Data model** – a vocabulary that is used to help diagram and outline how data is to be organized.

**Dataset** – a collection of data records.

**Domain** – in an ontology, a domain defines which classes a specific object or data property can be associated with.

**Information silos** – databases that are internal to an organization. The data within them can only be used and accessed within the organization and it is difficult, if not impossible, to share it with the outside community.

**Linked Data** – a method of publishing structured data that allows for maximizing interoperability. Linked Data is represented in the form of a triple.

**Domain specific vocabulary** – a vocabulary that is tailored for specific areas of interest.

**Open Data** – data that is free to be used and republished without any restrictions from patents or copyrights.

**OWL** – (Web Ontology Language) a W3C recommendation designed as an ontology language for the Semantic Web with formally defined meaning (referred to as a meta-vocabulary throughout this study).

**Protégé** – a software application that is used to build and maintain ontologies.

**Range** – in an ontology, a range restricts/specifies the types of objects that can be entered for that specific data property. For object properties, a range specifies which class/classes can be used as objects in the RDF Triple.

**RDF** – (Resource Description Framework) a standard grammar/syntax developed by the W3C for describing things on the Web.

**Semantic Web** – a collaborative movement led by the W3C to promote standard data formats on the World Wide Web

**SKOS** – (Simple Knowledge Organization System) a W3C recommendation designed for representation of structured controlled vocabularies.

**URI** – (Uniform Resource Identifier) a standardized string of characters used for identifying an abstract or physical thing

**XML and RDF prefixes** – prefixes that are used to contextualize an element or attribute within data. In RDF, prefixes are used to identify the specific vocabulary that a particular class, object property or data property is associated with. The prefixes and associated schemas/ontologies used in this study are listed below (Table 1).

Table 1: XML schemas and RDF ontologies used in this study

| XML prefix | XML schema |
|---|---|
| vra: | http://www.vraweb.org/vracore4.htm |
| xsd: | http://www.w3.org/2001/XMLSchema |

| RDF prefix | RDF ontology |
|---|---|
| schema: | http://schema.org/ |
| foaf: | http://xmlns.com/foaf/0.1/ |
| rdfs: | http://www.w3.org/2000/01/rdf-schema# |
| void: | http://rdfs.org/ns/void# |
| dcterms: | http://purl.org/dc/terms/ |
| skos: | http://www.w3.org/2009/08/skos-reference/skos.html# |
| vra-p:[1] | http://purl.org/vra/ |

**XML schema** – (Extensible Markup Language) a plan or model used for constraining XML records.

**XSLT stylesheet** – (Extensible Stylesheet Language Transformations) a document that is used to map XML documents into a new form. The mapping can be used to alter the data or simply to change its presentation.

[1] This is a proposed RDF ontology and prefix that was used throughout this study

# Chapter II

# Literature Review

## Linked Data Layer Cake

Linked Data is a method for publishing structured data. The format allows for maximizing interconnectivity between data. The links between Linked Data sets are designed to be leveraged in an effort to make the data more understandable and increase usability and interoperability. The concept behind the development of Linked Data is to add identity to data. Without identity, there is no way to determine reliability, and consequently no way to validate the authority of data found during an information search (Glaser & Halpin, 2012). While the W3C organization has overseen the development and implementation of Linked Data, the idea was first proposed by Tim Burners-Lee, who is widely credited with the creation of the Web (Burners-Lee, 2006). He proposed the use of URIs (Universal Resource Identifiers) not only to identify Web pages, but also to identify everything (both tangible and intangible). In order to use Linked Data on the Web, a structure was established to define how URIs could be used and how they would be resolved and connected with other URIs. An easy way to understand how the mechanics of Linked Data fit together is to visualize its structure as a layer cake (Figure 1):

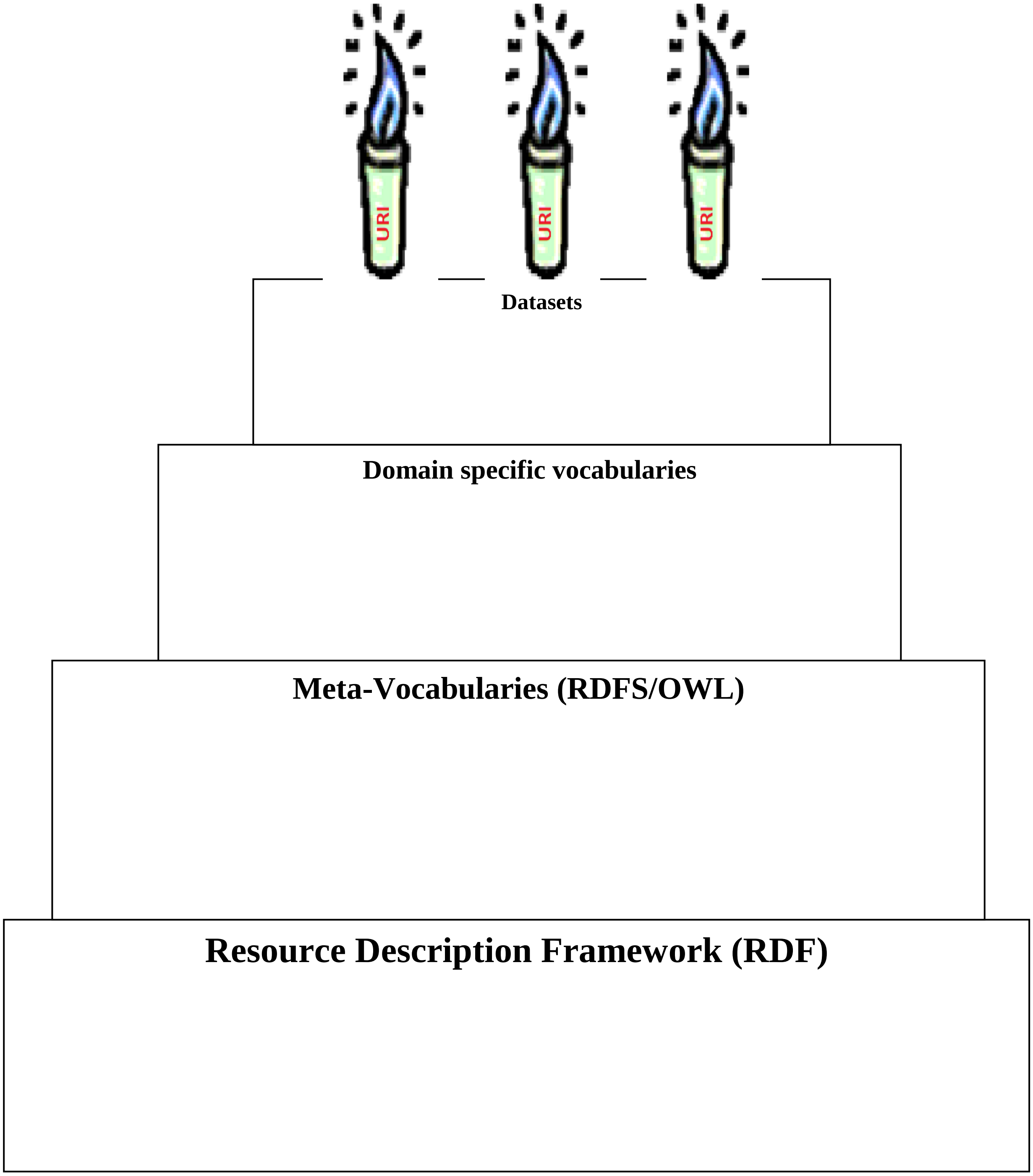

URI
URI
URI
Datasets
Domain specific vocabularies
Meta-Vocabularies (RDFS/OWL)
Resource Description Framework (RDF)

Figure 1. Linked Data Cake

### Resource Description Framework (RDF).

The description framework for Linked Data creation and implementation on the Semantic Web is called Resource Description Framework (RDF). RDF serves as an underlying structure upon which services can be built (Coyle, 2012). The basic architecture for the creation of RDF is very similar to the basic architecture employed in the development of a simple English sentence. The first term is a subject, the second is a predicate, and the third is an object. This sequence is called an RDF triple.

*Subject => Predicate => Object*

From this basic structure, schemas can be built, placed on top of the RDF structure and used to build complex ontologies to help in the structuring and organization of data.

### Meta-vocabularies (RDFS/OWL).

Meta-vocabularies are used to define and categorize the elements of the basic RDF triple. They also can be used to create mappings between two different domain specific vocabularies (discussed in section 2.2.3). Through the use of meta-vocabularies, one can begin to add meaning to the basic RDF triple.

*Subject = owl:Class or literal*

*Predicate = owl:ObjectProperty*

*Object = owl:Class*

Once these basic connections are made, an ontology or data model can begin to be developed. RDFS and OWL are the meta-vocabularies used in Linked Data. Meta-vocabularies establish the structure for domain specific vocabularies and establish the rules that domain specific vocabularies use to create detailed classes and properties.

#### ***OWL.***

OWL is a meta-vocabulary that can be used to create much more detailed relationships and also to develop and describe ontologies (W3C, 2004). There are various incarnations of OWL that can allow for refinement of more basic RDF vocabularies.  OWL serves as the basis for Protégé, a software application that can be used for creating ontologies.

### **Domain specific vocabularies.**

A domain specific vocabulary is used to define the specific elements of a dataset.  It acts as an overlay to the basic structure defined by RDF and is used to explain what the individual URIs identify and how they are related to other URIs.

#### ***SKOS.***

One of the simpler domain specific vocabularies is SKOS, which was published by the W3C itself.  It is used to define concepts and to establish simple taxonomical relations.  SKOS includes classes such as skos:Concept, and properties of concepts such as *broader, narrower, related* and *definition* .  An extensive list of SKOS elements has been published by the W3C.  All of these elements can be used better to define and explain a given URI.  A basic example of SKOS would be

a skos:Concept
*skos:preLabel "German Shepherd"*
*skos:broader"dogs"*

The example above, while very basic, explains that the thing being described is a concept and that the preferred term to identify the concept is "German Shepherd".  The last line indicates that the German Shepherd has the concept "dogs" as a broader concept.

### ***Schema.org.***

While SKOS can be used to describe things in general, it is not detailed enough to be used extensively in a library setting. In order to add the level of specificity required to describe specific things on the Web, Schema.org was developed (Coyle, 2012). The Schema.org vocabulary includes highly specific elements, referred to as properties, which can be used to describe diverse types of things. An example of this level of detail can be shown in how a librarian would go about describing a book. The item is a schema:Book which is under the category schema:CreativeWork which is grouped under schema:Thing. Within the schema:Book class there are various properties that take on values such as text, integers, URIs, or another types of schema:Thing. Google, Yahoo, Bing and Yandex collaborated in the development of Schema.org, with the idea that it would allow for a uniform way to describe things on the web. Although there has not been any empirical study to show how/if the search engines are using Schema.org markup in their search algorithms, Google recently has launched its Google Knowledge Graph, which aggregates information from the Semantic Web (Singhal, 2012). In order to allow for interoperability, multiple domain specific vocabularies can be used when describing an item. This allows people to leverage fully the power of Linked Data. In this way, a cataloger could use Schema.org to describe richly an item (in a similar way that MARC is used in library databases) while also using SKOS or other domain specific vocabularies to connect the thing being described with other things (in a similar way that thesauri do).

### **Datasets.**

The top layer of the Linked Data cake is RDF datasets. RDF datasets are similar to traditional controlled vocabularies, except that the name is not written out as a string of letters but rather as a URI. RDF datasets can be created or reused by any organization and serve as a reference point for the specific thing being referred to. DBpedia.org is a dataset that uses Wikipedia as a basis for its URI identifiers. Currently, the English version of DBpedia.org includes descriptions of 3.77 million things. Of these, 2.35 million are classified in a consistent ontology. The specific breakdown of the ontology is as follows: 764,000 persons, 573,000 places (of which 387,000 are populated places), 333,000 creative works (of which 112,000 are music albums, 72,000 are films and 18,000 are video games), 192,000 organizations (of which 45,000 are companies and 42,000 are educational institutions), 202,000 species and 5,500 diseases (http://wiki.dbpedia.org/Datasets). The library community was an early adopter of the Linked Data concept, and numerous library organizations have published their vocabularies as Linked Data datasets. In April 2009, the Library of Congress Subject Headings (LCSH) was published as Linked Data. OCLC published FAST (Facets Application of Subject Terminology) as Linked Data in 2011 (OCLC 2011). Other libraries, including the National Library of France, the National Library of Germany, and the National Agriculture Library, also have published their subject headings as Linked Data (Coyle, 2012).

### *Mapping datasets.*

As with domain specific vocabularies, datasets can be used interchangeably within Linked Data. This allows for an almost infinite amount of “controlled terms” and connections between terms. What makes this point even more important is that datasets can be meshed together very easily. For example, the unique URI used to describe “*dogs*” in LCSH can include a skos:exactMatch property with a link to the unique URI used to describe “*dogs*” in DBpedia.org. This must be done manually, and therefore is very time-consuming task. Additionally there can be problems relating to granularity and exact definition. The skos:exactMatch is very specific and indicates that the two terms are identical both in meaning and in use. This becomes difficult when trying to make connections between two distinct vocabularies. For example the URI for "*Mark Twain*" in LCSH might be used to describe works by the author Mark Twain, while the URI for "*Mark Twain*" in DBpedi.org might be used to describe the person Mark Twain. These issues generally require intellectual and even philosophical examination before they can be determined.

**RDF/XML.**

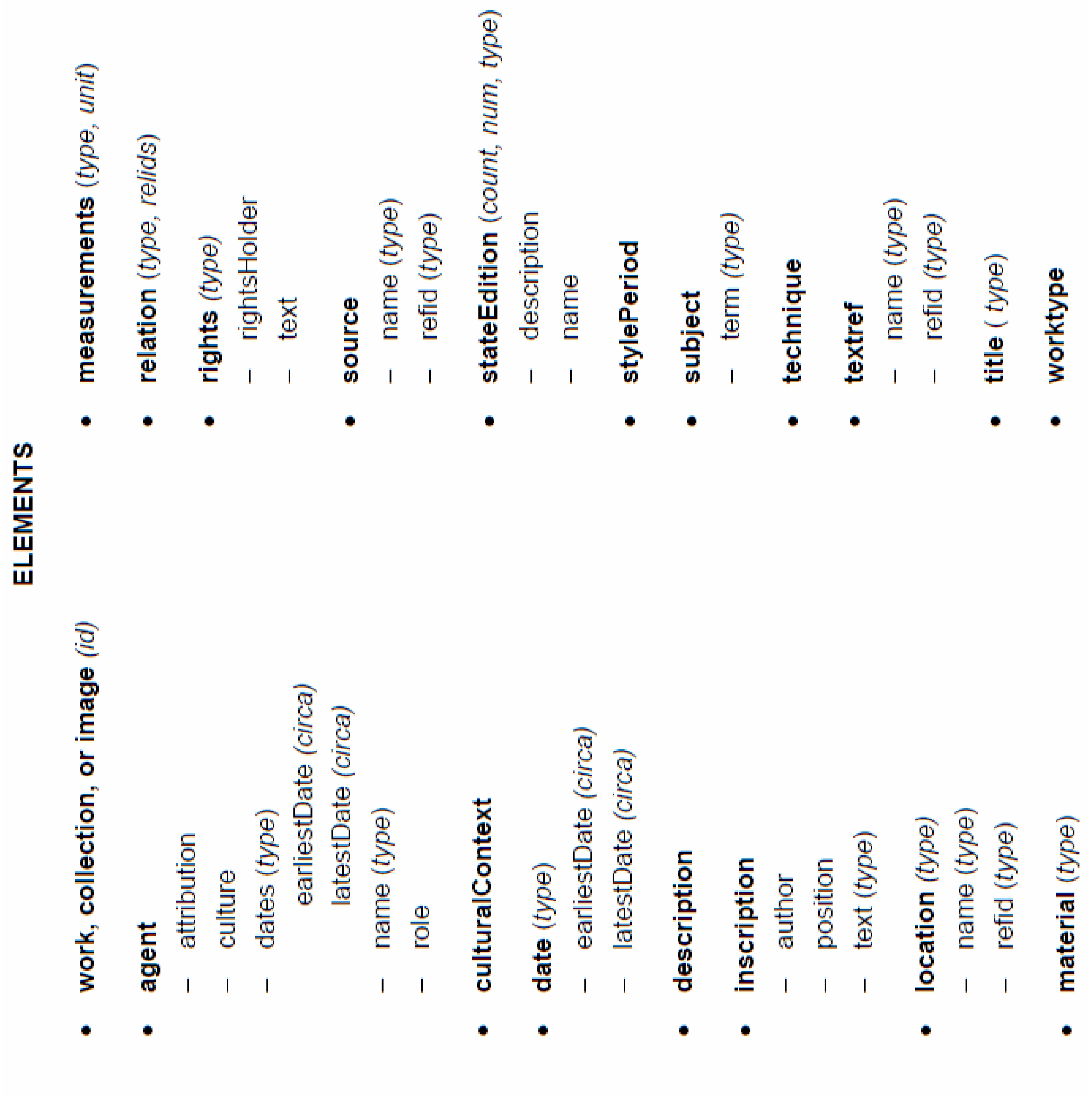
Figure 2 is an example of an RDF/XML from the W3 Schools Webpage. The record describes two albums, their associated artists, country of publication, publishing company, publication year and price. Although this example is rather simple, and can be improved by including URIs for applicable sting terms and using standard vocabularies, it nevertheless illustrates how RDF is rendered in XML. Regardless of whether one to one URI matching can be successfully implemented into the major vocabularies, the ability to use vocabularies interchangeably allows for a massive variety in unique and controlled terms. (Figure 2)

```
<?xml version="1.0"?>

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:cd="http://www.recshop.fake/cd#">

<rdf:Description
rdf:about="http://www.recshop.fake/cd/Empire Burlesque">
  <cd:artist>Bob Dylan</cd:artist>
  <cd:country>USA</cd:country>
  <cd:company>Columbia</cd:company>
  <cd:price>10.90</cd:price>
  <cd:year>1985</cd:year>
</rdf:Description>

<rdf:Description
rdf:about="http://www.recshop.fake/cd/Hide your heart">
  <cd:artist>Bonnie Tyler</cd:artist>
  <cd:country>UK</cd:country>
  <cd:company>CBS Records</cd:company>
  <cd:price>9.90</cd:price>
  <cd:year>1988</cd:year>
</rdf:Description>



</rdf:RDF>
```

Figure 2. Figure showing a Sample RDF. Source: W3Schools,
http://www.w3schools.com/rdf/rdf_example.asp

## VRA Core

VRA Core 4.0 was developed in an attempt better to include cultural-heritage data into metadata records. The new version of VRA was officially endorsed by the METS Editorial Board in 2007 (VRA Core, 2011). The primary difference between Core 3.0 and Core 4.0 was the division of records types into 3 distinct categories: Work, Image, and Collection. Previously there had been only Work and Image record types (LOC, 2007). The addition of the Collection category allowed for a more detailed description of the item or collection. Another major difference was the introduction of elements and attributes. The development and implementation of hierarchical description allows catalogers better to incorporate important cultural-heritage data (VRA, 2007). The new elements were developed in order to address issues that other visual image cataloging data models, such as CDWA (Categories for the Description of Works of Art) and CCO (Cataloging Cultural Objects), were beginning to address. The final major change was the ability to export VRA Core 4.0 records in flat XML files (LOC, 2007). This allows organizations to import pure VRA 4.0 XML files directly into services, such as CONTENTdm, that do not use relational databases as a foundation for record management (Carter, Double, Rose-Sander, & Webster, 2011). Although CONTENTdm does provide a template for VRA 3.0, the fields still are stored in a Dublin Core style record format, and for all intents and purposes the VRA 3.0 fields simply are superimposed over existing DCterms. Below is a figure detailing the VRA 4.0 elements, subelements and attributes (Figure 3):



**ELEMENTS**

- **work, collection, or image** *(id)*
- **agent**
  - attribution
  - culture
  - dates *(type)*
    - earliestDate *(circa)*
    - latestDate *(circa)*
  - name *(type)*
  - role
- **culturalContext**
- **date** *(type)*
  - earliestDate *(circa)*
  - latestDate *(circa)*
- **description**
- **inscription**
  - author
  - position
  - text *(type)*
- **location** *(type)*
  - name *(type)*
  - refid *(type)*
- **material** *(type)*
- **measurements** *(type, unit)*
- **relation** *(type, relids)*
- **rights** *(type)*
  - rightsHolder
  - text
- **source**
  - name *(type)*
  - refid *(type)*
- **stateEdition** *(count, num, type)*
  - description
  - name
- **stylePeriod**
- **subject**
  - term *(type)*
- **technique**
- **textref**
  - name *(type)*
  - refid *(type)*
- **title** ( *type)*
- **worktype**

Figure 3. VRA 4.0 Elements. Source: Library of Congress: http://www.loc.gov/standards/vracore/VRA_Core4_Outline.pdf

In a 2011 survey of 103 voluntary respondents, it was determined that 84 collections currently were using VRA Core data standard (see Figure 4). Of those, 56 were using Core 4.0; 25 were using Core 3.0; and 3 were using either Core 2.0 or 1.0 (Carter, Double, Rose-Sander, & Webster, 2011). It also was determined that the major deterrent towards not adopting VRA Core 4.0 was the "lack of understanding of how to implement the relational structure of the data model into local systems as well as a lack of institutional technical support for implementation" (Carter, Double, Rose-Sander, & Webster, 2011). The survey also asked respondents to comment on the extent to which their organizations, if applicable, used the VRA Core 4.0 elements, attributes, and record types. Of these, 42.3% responded that they had fully implemented elements, while 48.1% indicated that they had partially implemented elements. With regard to the implementation of attributes, the percentages were much lower. Only 19.2% indicated that they had fully implemented attributes, while 40.1% indicated that they had partially implemented attributes. The record types were used on average more frequently by organizations: 57.7% used Core record type = work; 53.8% used Core record type = image; and 25% used Core record type = collection (Carter, Double, Rose-Sander, & Webster, 2011).

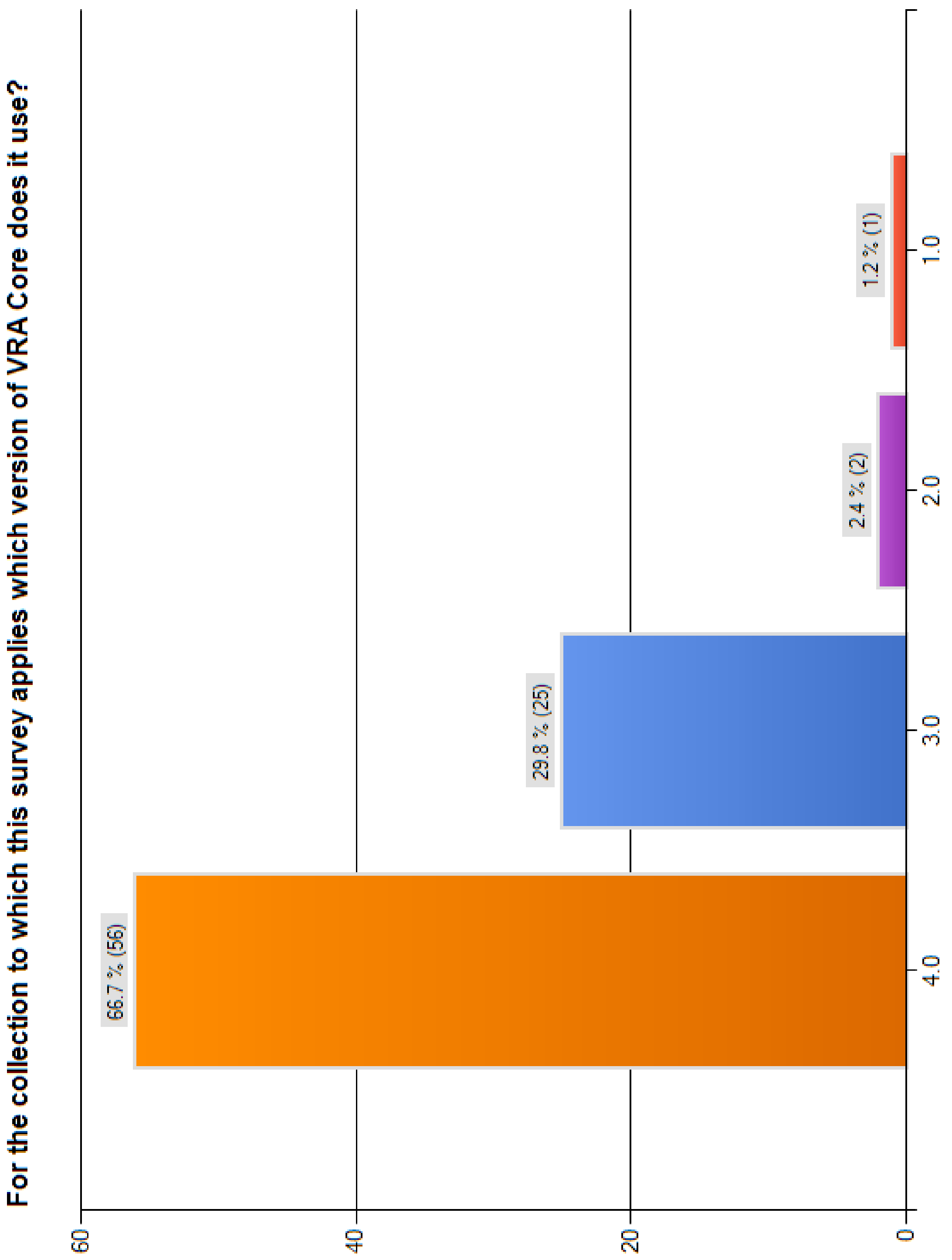


Figure 4. Survey Results from an interview regarding the use of VRA. Source: Carter, Double, Rose-Sander, & Webster, 2011.

## Initiatives in Publishing Linked Open Data

The recent push to publish data as Linked Open Data is based on the importance of connecting relevant data across the Internet. Once information, represented as RDF, is published using Linked Data principles, it becomes possible to create links to and from other related or relevant datasets. One of the main reasons for publishing data in a standard RDF format is to help overcome the problem of sharing information across incompatible formats. Traditionally metadata has been stored in spreadsheets, databases and unique file formats (including MARC). Even though humans desire to share information, desktop tools by design are not optimized for interoperability (Miller & Westfall, 2011). Furthermore, the lack of interoperability not only hinders information sharing, but also prevents data portability. In the past, computer application had to be used to create raw data and subsequently share it with others. Publishing data in an RDF format using Linked Data principles can be used as a way to bridge the gap between the raw data and be able to share it with the world (Miller & Westfall, 2011).

In 2009, President Obama announced that government organizations would begin to publish data as Linked Open Data (Miller & Westfall, 2011). The strategies that were developed to support these initiatives included leveraging open formatted data, developing best practices for data, and linking open data semantics and descriptive metadata. Eric Miller and Micheline Westfall have pointed out that "[t]hese are all incentives that libraries have been discussing and developing for a much longer period of time" (Miller & Westfall, 2011). In addition to the United States government, other organizations have begun to publish Linked Open Data. The BBC uses is own published Linked Data to generate rich interconnected websites. Through the use of Linked Data, the BBC is able automatically to link related articles regardless of topic. Although this was possible in the past, the process was not scalable and required humans to manually create the links between related articles (W3C, 2012a). By comparison, the BBC now has a robust semantic information base that automatically generates links to related topics. Even more importantly, since the process is automatic and based on a continually evolving/growing open Linked Data repository, the system also is capable of continually updating itself by adding relevant topics/articles as they develop and removing outdated ones (W3C, 2012a). In order for organizations to share data more efficiently, and eventually to use and implement that shared data, it should be published as Linked Open Data using vocabularies that are understandable by the larger community.

After data has been published to the Semantic Web, it then can be connected to related data. Additionally, the data can be aggregated and used to form complex and comprehensive information resources. Once the information resources are developed, they then can be used to create data mash-ups that in turn can be used to create tools and resources. Freebase (http://www.freebase.com/) is one organization that is aggregating information from the Semantic Web, including DBpedia.org, and organizing it into a comprehensive information resource (Coyle, 2012). Google uses Freebase as a primary data resource for creating and populating the information included in the Google Knowledge Graph (Singhal, 2012). Information from across the Semantic Web is pulled together to create a rich source of structured, traceable and identifiable data. For example, the Freebase entry for *Mark Twain* (http://www.freebase.com/view/en/mark_twain) aggregates information from 15 different datasets, including VIAF, Open Library, the Library of Congress Name Authority File and Wikipedia that are represented on the Semantic Web. This type of information aggregation is possible only if organizations make the effort first to publish Linked Open Data on the Semantic Web.

# Chapter III

# Methodology

## Overview

This study began by reviewing the current VRA Core 4.0 restricted XML schema. This schema was used as a template for developing a new VRA data model and the ontology. It was decided that a XSLT stylesheet would be developed as an instrument to convert XML records into other formats. Typically XSLT is used for converting XML into HTML, but for the purpose of this study the stylesheet instead was used to convert the existing VRA XML records into RDF. Reference data provided from the LOC/VRA website initially was used better to understand how VRA records were constructed and how they were formatted in XML. This was important because the XSLT stylesheet was designed to identify specific XML tags and convert them into RDF. In order to make the stylesheet as detailed as possible, records from a research sample were then employed as use cases to supplement the reference data provided by the LOC/VRA website. Below is a list of specific tasks that were conducted during the study (more detail on these tasks is provided in section 3.2):

1. Develop a data model
2. Develop an ontology
3. Design a XSLT stylesheet
4. Test and refine the stylesheet using sample data

## Research Design

### Developing a data model.

#### *Universal modeling language.*

The first task in the study was to develop a new data model to support the development of the ontology and guide the creation of the XSLT stylesheet. Enterprise Architect, a UML (Universal Modeling Language) tool was used to develop the new VRA data model. In addition to being used as a reference for later tasks, the data model also helped visualize how existing domain specific vocabularies and custom VRA terms could be combined together to form a coherent and semantically rich representation of the existing VRA Core 4.0 model.

#### *Schema.org.*

Schema.org was used as the preferred domain specific vocabulary. As such, it was used to map the XML specific elements and attributes of the VRA Core 4.0 data model into RDF elements and properties. Other visual resource data models, such as CIDOC, chose to create custom RDF elements and properties (CIDOC, 2009 February 23). Using an entirely custom RDF vocabulary poses future problems with regards to implementation, adoption and audience. Most troubling, it violates one of the founding rules of Linked Data as prescribed by the W3C, which is to use standard RDF (Berners-Lee, 2009). In order for Linked Data to be useful, it needs to be connected to other similar data. This is easily achieved if popular vocabularies are used to describe information. If data is not connected to other data, one is left with a dataset that is an isolated island within the ever-growing interconnected archipelago of the Web.

### Developing an ontology based on the data model.

#### *Resources used to develop the ontology.*

The second task of the study involved developing an ideal schema that could be used to convert existing VRA 4.0 records into RDF. A preliminary ontology was developed based on the existing VRA 4.0 restricted XML schema. In order better to understand how this schema was reflected in existing VRA records, sample data provided by the VRA 4.0 website (http://www.vraweb.org/projects/vracore4/) were used as preliminary case studies. One of the initial problems encountered in this task was the fact that the VRA restricted XML schema is a suggested model and there are very few requirements for records to validate against the schema. Additionally, the requirements are primarily focused on high-level/broad elements and attributes. When deciding where to include controlled vocabulary headings, there are very few requirements. Rather, VRA 4.0 only suggests recommended controlled vocabularies to use. Additionally, there are no mechanisms/checks in place to validate if a cataloger has chosen a recommended vocabulary or if the vocabulary ID used is correct.

Since the existing VRA 4.0 data model is vague on specific requirements and very open for interpretation and implementation, it was decided that the best way to continue with the study was to use an existing VRA 4.0 sample collection from Notre Dame to develop the ontology and the XSLT stylesheet. The sample was reviewed and determined to be of a high enough quality to use as a basis for developing the ontology and the XSLT stylesheet (see section 3.3: Research Sample for more detailed information on the sample set).

#### *Creating ontological classes.*

Using the Notre Dame sample set for reference, a detailed ontology was developed using Protégé. The development of the RDF classes was relatively simple, and many of the existing

VRA XML elements easily were converted directly into Schema.org classes. In some cases, other vocabularies had to be used to aid in the translation of VRA into Schema.org. One such instance was the vra:agent element. Schema.org does not have a generic Agent class, but it does have both Person and Organization classes. The FOAF (Friend of a Friend) vocabulary, which has an "Agent" class, was used in this instance, and the Schema.org Person and Organization classes then were used a sub-classes to add even more semantic detail to the model.

One problem in converting the sample set records was determining how to properly deal with the collection record. This issue speaks to one of the fundamental differences between traditional cataloging and RDF. Catalogers create records about things, while RDF is used directly to describe things in that exist in the world. Using RDF, a library database would not be a collection of records about items located in the library, but rather a list of triples describing various entities (for example a place or an author). When combined with other descriptions (i.e. triples) a compound description can be formed (similar to a traditional metadata record), but in essence this is nothing more than a group of triples that otherwise are independent. In the case of the sample set, the collection record was not a thing that could be described; rather, it was a record describing the collective whole of the sample set. In order properly to address this problem, the VoID data model (http://www.w3.org/TR/void/) was selected. VoID was specifically designed to describe abstract concepts, such as data sets. A void:DataSet is defined as a meaningful collection of triples that are aggregated together for a specific purpose. Since the vra:collection in the Notre Dame sample set is used to describe what is in the collection, the void:DataSet class seemed an appropriate match in regards both to definition and use.

#### ***Creating ontological properties.***

Since the VRA 4.0 data model was designed to be used in a XML schema, there were various problems in creating ontological properties. In order to add detail to the new VRA RDF data model, existing VRA XML elements needed to be parsed and converted into new ontological object properties. This problem was most evident in attempting to convert the VRA date element into the new VRA RDF data model. In the existing XML schema, the vra:date element can be modified with various type attributes (for example “creation”). This combination is easily understood by a human to represent the dates (i.e. event) during which the object being described was created. Unfortunately, a machine cannot logically combine the individual pieces of the XML element to develop the same relationship between the XML element and the object being described. In order to create a complex machine-understandable data model, while still retaining the human-understandable complexity of the existing XML schema, all of the date element and type attribute combinations had to be individually mapped into new ontological object properties. Adding to this complexity is the issue that, in some instances, the vra:date element represents an ontological data property. When the vra:date element is combined with the “life” type attribute, the semantic meaning of the element changes from an object property, described above, to an data property simple defining the life dates of a vra:agent. In order to effectively map the VRA restricted schema into a new data model, both the XML elements and element-attribute combinations needed to be understood and addressed.

#### ***Creating custom VRA elements.***

In the event that any of the VRA XML elements did not match any existing RDF vocabularies, custom VRA classes were created. These classes were labeled with the prefix “vra-p” (VRA proposed) and were posted to a PURL (Persistent uniform resource locator) website (http://purl.org/vra/) so that they could be understood both by machines reading the RDF code and by humans interested in understanding how to use the ontology. This pattern of creating custom VRA RDF tags also was used when creating object properties as well as data properties. Whenever possible, the custom VRA RDF tags were made sub-classes or sub-properties of existing Schema.org classes and properties. This practice would allow both the Schema.org tags as well as the custom tags to be generated in the RDF output. Since the major search engines understand and consume Schema.org, including both the broader Schema.org classes/properties and the custom VRA classes/properties helped to maximize the visibility of the data. It is also anecdotally assumed that this practice can improve the SEO (Search Engine Optimization) of websites. If the custom VRA classes and properties are used in future VRA data creation, an OWL Reasoner can be used to help infer and create these relationships. Since the custom VRA ontology has not been adopted or approved by the VRA 4.0 board, the relationships were manually coded into the XSLT stylesheet.

### *Adding semantics to the ontology.*

The final step in creating the VRA ontology was assigning domains and ranges to all of the object properties and data properties. This was complicated by the fact that the existing VRA XML schema uses many “free floating” attributes that can be applied any element. In most cases the “free floating” attributes were eliminated because there were split into distinct Schema.org data properties. For example, the VRA attribute “earliestDate” and “latestDate”, which can be used under both the vra:agent and vra:date elements, were divided into schema:birthDate and schema:deathDate as well as schema:startDate and schema:endDate. The schema:birthDate and schema:deathDate data properties were given domains of the schema:Person class, while the schema:startDate and schema:endDate data properties were given domains of Schema.org

"Event" class. All of these date related data properties were given ranges of xsd:date, which is a W3C standard format for expressing dates (http://www.w3schools.com/schema/schema_dtypes_date.asp). In this instance a date must be entered as YYY-MM-DD. Applying specific ranges, such as xsd:date, prevents catalogers from using inconsistent data formats and allows machines to parse and understand the data in the RDF output.

### Designing an XSLT Stylesheet.

#### *Resources used to develop the stylesheet.*

The next task of the study was developing the XSLT stylesheet. Again, the research sample set was used to help guide the development of the stylesheet. This was a rather complicated process requiring a fair amount of trial and error in order to make sure that it worked properly. As with the development of the ontology, the Notre Dame sample set of records was used as the primary reference for building the stylesheet. Even though the sample set was used as the source for developing the stylesheet, the VRA restricted XML schema also was used to ensure that all of the possible VRA attributes were incorporated, regardless of whether they were used in the sample data or not. This strategy helped ensure that all of the VRA restricted schema elements and attributes were incorporated, while additionally tailoring the specifics of the XSLT stylesheet to work with the Notre Dame sample set.

#### *Converting controlled vocabulary terms into URIs.*

A key aspect of creating Linked Data is adopting and using URIs (Uniform Resource Identifier). While the preferred URI source is not as important as is the domain specific vocabulary, it is important to note that there currently is not a high level of interoperability between individual URI sources. In developing the stylesheet, it was important to retain as much valuable data from the original XML records as possible. In order to do this, custom script was developed to help parse the XML data and make valuable semantic connections to existing URIs.

Since the Library of Congress had published their LC vocabularies as Linked Data, it was possible to design XSLT scripts that converted all LCSAF (Library of Congress Subject Authority File), LCNAF (Library of Congress Name Authority File), LCTGM (Library of Congress Thesaurus for Graphic Material) and LCSH (Library of Congress Subject Headings) heading IDs into unique URIs (see Appendix B). Other vocabularies frequently used in VRA records and consequently familiar to image catalogers, such as Art and Architecture Thesaurus (AAT), Union List of Artist Names (ULAN), and Getty Thesaurus of Geographic Names (TGN), have not yet been published as Linked Data vocabularies (Isaac, Waites, Young, & Zeng, 2011). ULAN has been adopted and implemented in Virtual International Authority File (VIAF), and consequently VIAF URIs were used for all terms derived from the ULAN vocabulary. For the other Getty vocabularies (AAT and TGN), recommended URIs were constructed in this study. Since the Getty vocabularies have not yet been published as Linked Data, the stylesheet was designed to create URIs that could serve as examples for how Getty could publish its vocabularies (see Appendix C). As with the custom vra-p ontology classes and properties, these URIs were linked to a PURL website to demonstrate the importance of using links that are not subject to change or obsolescence.

In order to develop a stylesheet that could adapt to variety of different vocabularies, it was necessary to spend an extensive amount of time studying the actual Notre Dame sample set and determining exactly what vocabularies were used. Since the VRA Core 4.0 documentation provides only suggested vocabularies that the cataloger of the sample set frequently used vocabularies that were not included in the VRA recommendations (these included all of the published Library of Congress vocabularies). As a result, the stylesheet had to be customized to meet the needs of the corresponding XML sample set.

### ***Minimizing data lose.***

In the attempt to retain as much valuable data as possible during the conversion process from XML to RDF, there was a need to analyze carefully the sample set in order to determine where the valuable data was stored. In most cases, the data was stored within XML tags associated with a specific element:

```
<stylePeriod vocab="LCSAF" refid="sh 85091984">Nineteenth century</stylePeriod>
<stylePeriod vocab="LCSAF" refid="sh 85139020">Twentieth century</stylePeriod>
```

In some instances however, the data was not stored in the associated XML tag but rather in the "display" tag:

```
<materialSet>
        <display>marble</display>
        <notes />
        <material />
</materialSet>
```

In this instance it was necessary to pull the data directly out of the "display" tag. In most cases the "display" tag is used to specify what data is extracted and displayed to an end-user who might be viewing the item described by the record. In some cases however, the Notre Dame sample set used the display tag as a default space to code valuable data. If the stylesheet had been set up only to convert data from the actual XML element (in the case above the "material" element), the resulting RDF would have been empty. In order to avoid problems such as this, the stylesheet was designed to check for blank XML nodes, and in the event one was found to use the "display" value instead (see Appendix D).

### *Adding semantics to the data.*

In addition to customizing the XSLT stylesheet to ensure that all of the important record data is converted into RDF, it also was attempted to add additional semantic depth to the data. In order to achieve this, individual XML attributes were parsed out and used to create new RDF entities. One such instance in which this technique was used was during the conversion of the vra:date elements. This element has 15 associated type attributes for the data being described. For instance, a vra:date element could have a vra:broadcast type attribute that would signify that the dates in the description are related to when the item originally was broadcast. Since this is a relatively flat way to describe the item and it requires a human to decipher the meaning/semantics of the description, it was decided to turn each of these type attributes into object properties that connect the item being described with a schema:Event class. As a result, the RDF output data would describe that the item (for example a television show) is related to (via an object property called vra-p:wasBroadcast) a schema:Event (called Broadcast) and that this schema:Event has a start and end date. Describing an item in this way allows for a more complete description of the material and allows a machine to read and understand the semantics of the data.

### *Eliminating unnecessary data.*

The final task in developing the XLST stylesheet was to adequately deal with all of the original XML data that was not needed in the RDF. In general, this was data that was included in "display" tags as human readable string values. The reason that this information was not needed is that it does not add any value to the RDF:

```
<stylePeriodSet>
        <display>Renaissance</display>
        <stylePeriod vocab="AAT" refid="300021140">Renaissance</stylePeriod>
</stylePeriodSet>
```

In the example above, there is no value in converting the "display" value when the information also is available in the actual vra:stylePeriod element. This process of ignoring the "display" values was not implemented when the associated VRA element was blank. In the output RDF, data that was not converted by the stylesheet was commented out and therefore ignored by any application that was reading the RDF/XML.

### Testing and Refining the Stylesheet using Sample Data.

#### *Testing new templates.*

Since the stylesheet was rather long (consisting of approximately 2400 lines of script), there was a constant testing, reviewing and editing process as it was being developed in order to ensure that any changes or additions to the stylesheet did not cause problems with any previous scripts. Three records (a Collection record, a Work record and an Image record) were selected from the Notre Dame sample set. The three types of records in the Notre Dame sample all use the same cataloging format and style. The Collection record, Work record and Image record templates are similar throughout the entire collection (the only difference is obviously the data used to fill the records). Consequently, it was determined that for the purpose of this study a single record from the three different record types would be adequate for testing and refining the stylesheet as it was being developed.

In order adequately to cover the entire scope of the VRA 4 restricted schema, the records chosen for the sub-sample were selected based on how many of the restricted schema elements were included in the records. Since not all of the elements in the restricted schema were used in the sample set, measures were taken to ensure that these elements nevertheless were included in the XSLT stylesheet. Each of the schema elements was searched for individually within the sample set in order to find applicable use case data, which then could be used to help create an XSLT template. If no use case data was found, samples provided by the VRA website were used to help construct the appropriate XSLT template.

These three records were used to evaluate how each new template affected the overall XML conversion process. Although the three test records all came from the same sample set, there were differences in how the specific type of records were cataloged. The Work and Image records all had a “relation” element that is used to connect Works to other Works or Images to Works. Additionally, in this sample set the “measurements” elements in the Collection and Work records were formatted differently than the “measurements” element in Image records. In Image records, the cataloger used only the “display” tag to encode information, whereas in the Collection and Work records the values were coded with specific measurement types (i.e. inches, feet, etc.). In order to account for the minor difference in cataloging practices between the different types of records in the sample set, continual testing was done to make sure that all of the appropriate XML elements and attributes were being converted accurately and that no valuable information was being misrepresented in the RDF output.

### ***Back-checking stylesheet changes.***

In addition to making sure that the various XSLT templates work properly in converting the XML into RDF, the continual testing also was used to make sure that any changes or modifications made to one stylesheet template did not adversely affect other templates. Since the stylesheet consisted of 42 individual templates and eight sub-templates, it was important to confirm that they all work together. Even a minor change to a template could cause other templates or related sub-templates not to work properly when run against the sample XML records. In order to determine if a change to the XSL stylesheet caused any problems in the conversion process, a manually created expected outcome RDF/XML document was created. This document then was compared with the RDF/XML that was generated using the XSLT stylesheet conversion. Through identifying the differences between the expected outcome XML and the generated XML, any individual XSLT stylesheet templates that either were not working properly or not working at all could be identified, evaluated and fixed.

#### *RDF validation and manual review.*

As a final check to make sure that the XSLT stylesheet was working properly and generating the correct RDF, the RDF/XML output was run through the W3C RDF validation service (http://www.w3.org/RDF/Validator/ ). Although this service is not able to evaluate the quality of the RDF, it nevertheless was an important first step in determining how successful the XSLT stylesheet was at converting XML in RDF. The completed stylesheet was complete, was run against the entire 4,150 record Notre Dame sample set. The output then was manually reviewed in order to ensure that the XSLT templates worked and produced meaningful RDF.

## Research Sample

In order to obtain a quality VRA 4.0 sample, a request was sent out over the VRA listserv asking for assistance in obtaining a VRA 4.0 record sample that was already in a XML format. In response, Susan Jane Williams, a member of the VRA Data Standards Committee, provided all of the metadata records from the University of Notre Dame Lantern Slide collection (http://www.flickr.com/photos/ndalls/sets/72157605061425911/). The Notre Dame sample set was selected both because of its availability and because it was created and natively stored in XML.

### VRA population.

Taking a true statistical sample was difficult because there is no reliable way to determine the total population of VRA 4.0 records. Additionally, since VRA 4.0 is a suggested data model rather than a rigid metadata format (such as MARC21), it is difficult to determine the quality of the entire population of VRA. The problems associated with determining an accurate VRA 4.0 population size were confirmed by members of the VRA listserv community. Since this stylesheet was based on the restricted VRA 4.0 data model, it was important that the sample records were of high quality and conformed to the existing restricted XML schema. Since the stylesheet was tailored specifically to the restricted schema, any custom elements or attributes that catalogers might have added to the records were ignored.

### Sample size.

The sample contained 4,150 VRA 4.0 records in XML format. Using Creative Research System's Sample Size Calculator (http://www.surveysystem.com/sscalc.htm#one), it was determined that the confidence interval of using the Notre Dame sample of 4,150 records was 0.3. This calculation was based on a confidence level of 95%, an undetermined population size (calculated as an infinite population size) and a percentage variable of 99 (based on the fact that every sample record is a VRA restricted schema validated record).

### Quality of the Sample Set.

#### *Assessing the sample set.*

Before the sample set was selected for the study, all of the records were reviewed in order to ensure that they met the requirements of the VRA Data Standards Committee (http://vraweb.org/organization/committees/datastandards/). All of the records in the sample set validated against the current VRA 4.0 restricted XML schema, and all of the individual records also included the VRA elements required to provide "meaningful retrieval" (VRA 2007). For a Work record the required elements are: vra:workType, vra:title, vra:agent, vra:location and vra:date. For Image records the required elements are: vra:workType and vra:title (VRA 2007). After ensuring that the records in the sample set met the minimum requirements proposed by the VRA Data Standards Committee, a further review was conducted to determine the quality of the records. Since the existing VRA 4.0 schema is a suggested data model for crating VRA records, it was decided that the best way to determine quality was to compare the sample set against the VRA restricted schema. When comparing the sample set to the restricted schema both the quantity of restricted schema elements as well the quality of restricted schema elements were taken into consideration. After the assessment of the records, it was determined that the sample set was "clean", in that there were no major cataloging mistakes and the VRA 4 restricted schema was followed and implemented to the fullest extent.

### ***Quantity of VRA restricted schema elements.***

The sample set was very thorough in including all of the recommended restricted schema elements that were applicable. Of the 21 elements included in the restricted schema (http://www.loc.gov/standards/vracore/VRA_Core4_Restricted_schema_type_values.pdf), the sample set used 17 elements. Of the four elements that were not used (vra:textref (refid), vra:textref (name), vra:textref and vra:stateEdition), it was determined that none of them were applicable to the collection being described in the records and consequently were not needed.

### ***Quality of the VRA restricted schema elements.***

The quality of the elements was based on how frequently they were accompanied by refid and vocab values, which are used to link the string value to a controlled vocabulary. This decision was made based on the fact that, during the XSLT conversion, vocabulary refids are used to form the URIs that are generated in the RDF output. Since the URIs are vital in describing entities in RDF, and consequently vital in adding semantic richness to the sample set RDF output, it was determined that for the purposes of this study the quality of the sample set could be accurately judged through the use of controlled vocabulary terms that had accompanying refids. From the Notre Dame sample set the Agent, Location, StylePeriod, Subject, Technique and WorkType VRA 4.0 elements all had identifiers that connected the string values to controlled vocabularies. The vocabularies that were used were primarily ULAN, AAT and TGN, but LCSH also was used frequently for the Subject and StylePeriod elements. An example of such an attribute can be seen below:

```
<term type="descriptiveTopic" vocab="LCSAF" refid="sh 85040989">Education</term>
```

In reviewing the sample set, it also was discovered that some elements contained enough descriptive attributes that the XSLT could be designed to generate RDF machine readable structured values. Since a key aspect in RDF is creating data that a machine can understand, both with regard to structure and even more importantly with regard to meaning/context, it was determined that the use of structured values also was important in assessing the quality of the sample set. The Measurements element used enough attributes that the string values could be parsed and converted into machine-readable RDF triples. Below is a sample of a Measurements element that can be converted into machine-readable RDG triples:

```
<measurements type="height" unit="in">68.5</measurements>
```

# Chapter IV

# Results

## Overview

The results for the study clearly showed that it is possible to convert flat XML data into rich RDF data using an XSLT stylesheet. Even though the XSLT stylesheet required some unique customization to work effectively with the sample data, the VRA ontology that was developed during the study remained consistent and can be used/applied by any organization with VRA data, assuming that it conforms to the existing VRA restricted schema. Finally, this study confirmed that even complex data models such as VRA Core 4.0, that were designed with the intention of being implemented in flat XML formats, can be translated into RDF without developing an entirely new ontological model. This is important because it reinforces the founding principle of Linked Data, as described by the W3, and provides organizations with the opportunity to make their data more visible and connected.

## Data Model

### Overview.

The data model that was developed for the study reflected how existing vocabularies could be combined with custom elements to form a semantically correct description of visual items. The model served as a basis for the later development of the ontology, and provided a visual representation of how the existing VRA records would be mapped and converted into RDF.

### UML model.

In the following figure a UML (Unified Modeling Language) model is presented. UML is a standardized modeling language and is widely used in data modeling. The class diagram gives a structural view of the major classes and their relationships (Figure 5).  For example, in the center is the CreativeWorks class, which is related with several other classes such as Agent, Material, Technique, etc.  The classes and relationships are reflected in the ontology to be described in the following section.

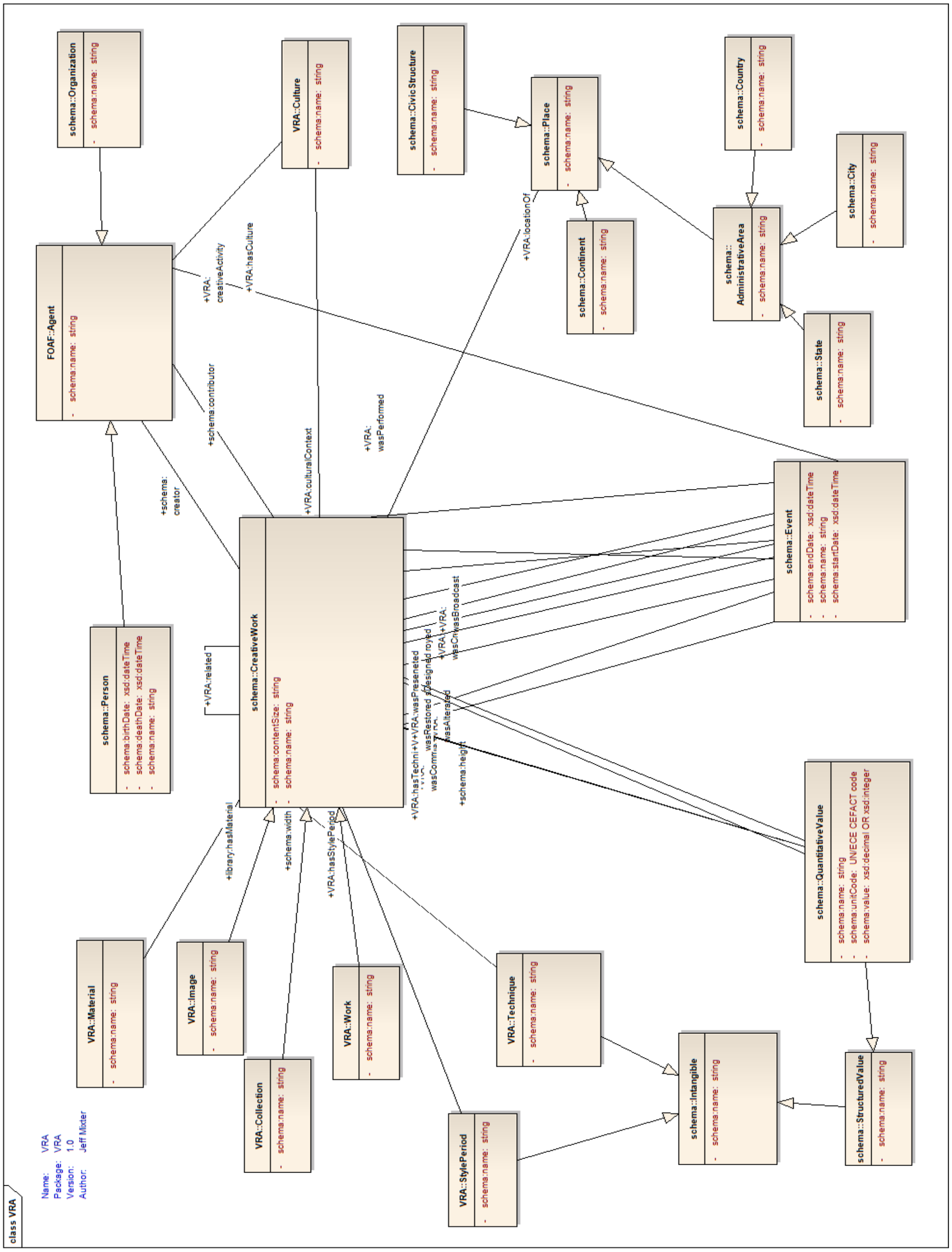


Figure 5. Data Model developed for the study

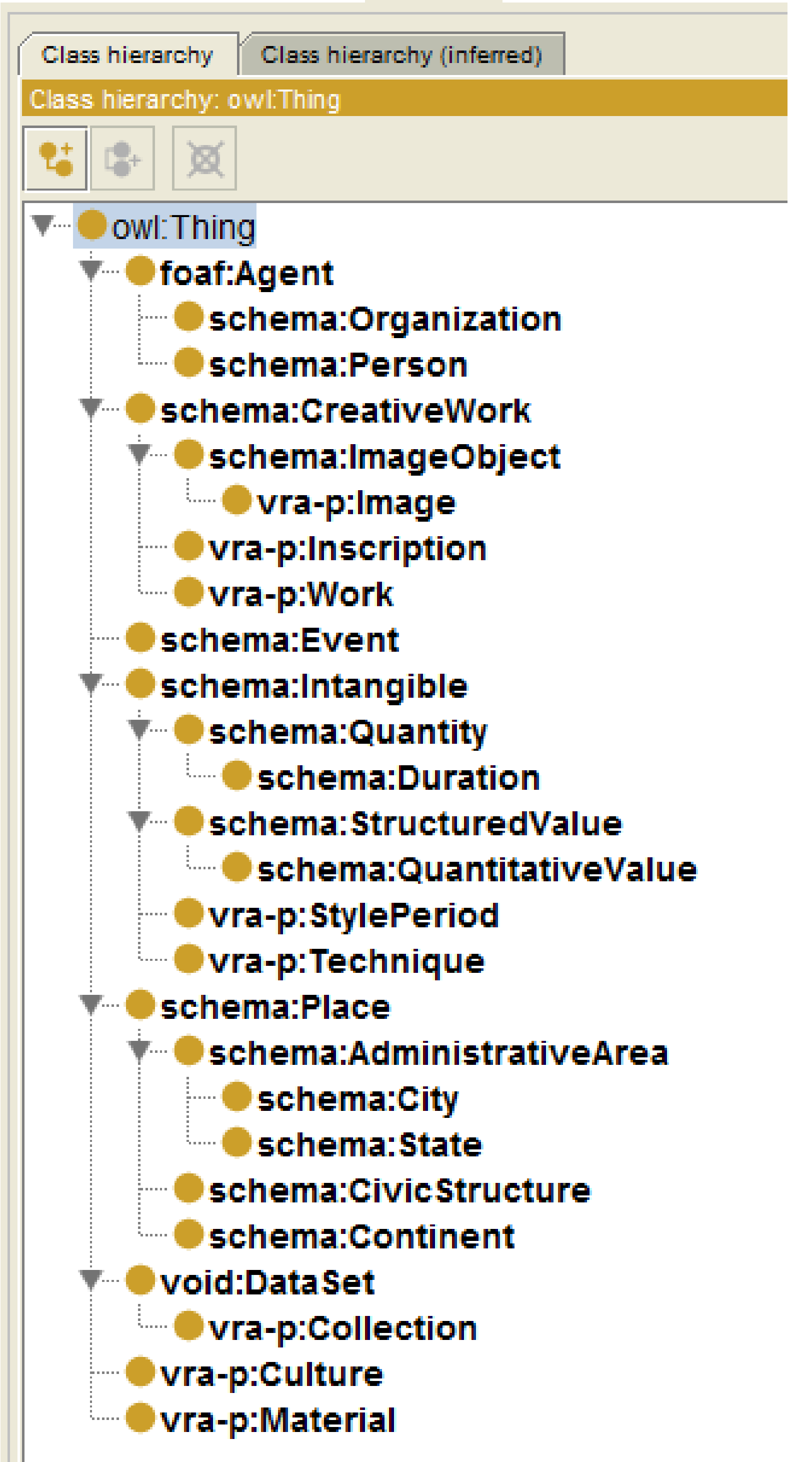
## Ontology

### Overview.

The resulting ontology from this study used 35 Schema.org classes or properties, one FOAF class, one Dublin Core Terms property and two VoID classes or properties, and also required the use of 127 custom VRA classes or properties. Of those 127 custom VRA classes, six were used as subclasses of existing Schema.org classes and 78 were used as sub-properties of existing Schema.org properties. While the raw statistics show that there were more custom VRA classes and properties used, it is important to examine carefully exactly how and why the custom properties and classes were used. Since the VRA 4.0 is a very detailed data model for cataloging works of visual culture, there were many highly specific relationships that simply could not be accounted for by using just Schema.org and FOAF. In order to minimize data loss and to retain as much detail as possible, it was necessary to include all of the highly specific VRA XML attributes as custom object properties.

### Classes.

The classes in the VRA ontology reflected very well the existing VRA XML schema. There are eight main classes, such as foaf:Agent, schema:CreativeWork and schema:Place. As illustrated in the following figure, appropriate classes from namespaces such as FOAF and Schema.org were used whenever possible, such as foaf:Agent. Of the 24 total classes, only eight were custom VRA classes which were designated with the prefix “vra-p”. Additionally, six of those eight were sub-classes of existing popular domain specific vocabularies (i.e. FOAF or Schema.org). The entire VRA Ontology can be viewed/downloaded in RDF/XML form as well as HTML form from http://purl.org/jmixter/thesis/, and a screenshot of some of the classes is presented in Figure 6 below.

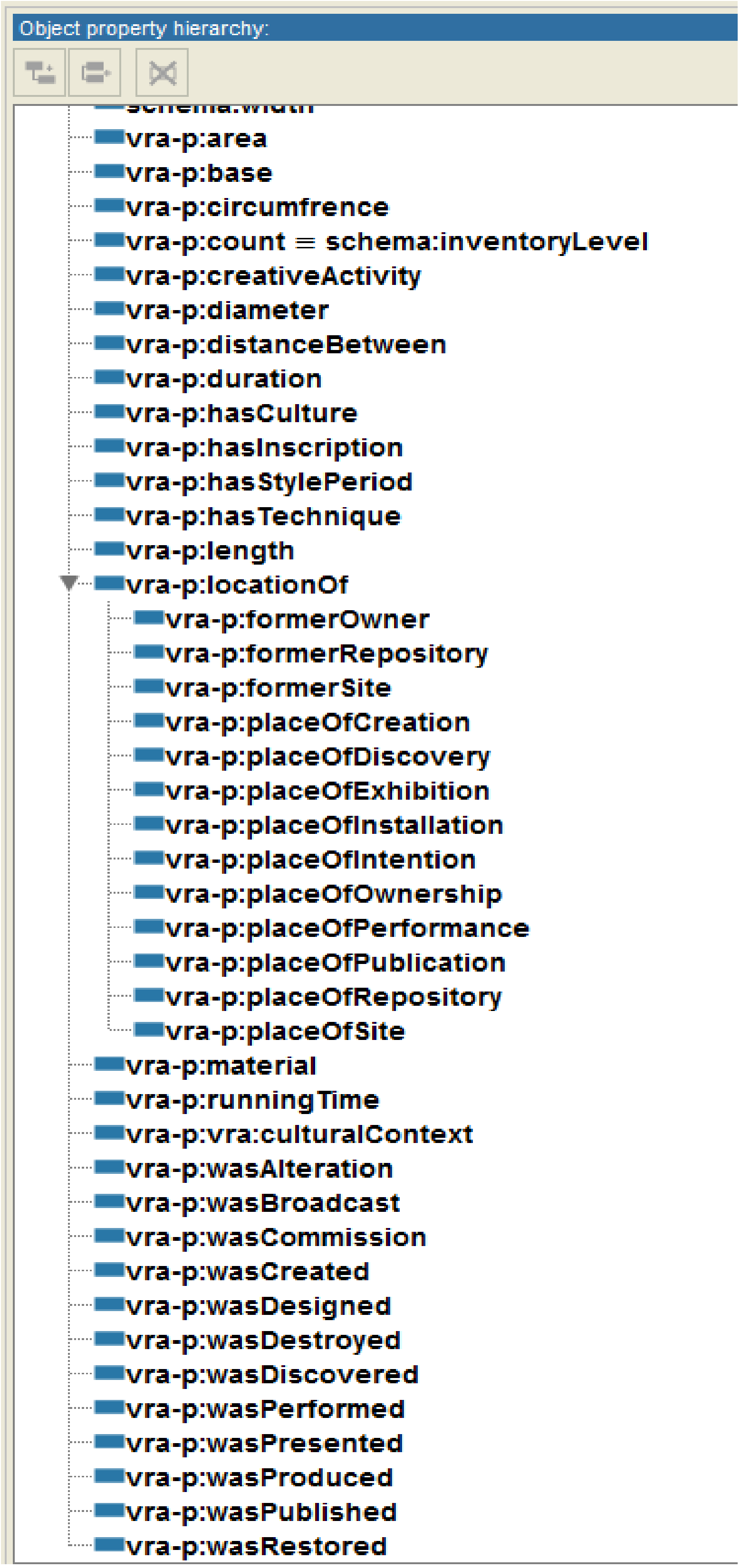


- owl:Thing
  - foaf:Agent
    - schema:Organization
    - schema:Person
  - schema:CreativeWork
    - schema:ImageObject
      - vra-p:Image
    - vra-p:Inscription
    - vra-p:Work
  - schema:Event
  - schema:Intangible
    - schema:Quantity
      - schema:Duration
    - schema:StructuredValue
      - schema:QuantitativeValue
    - vra-p:StylePeriod
    - vra-p:Technique
  - schema:Place
    - schema:AdministrativeArea
      - schema:City
      - schema:State
    - schema:CivicStructure
    - schema:Continent
  - void:DataSet
    - vra-p:Collection
  - vra-p:Culture
  - vra-p:Material

Figure 6.  Classes in the VRA ontology

### Object properties.

One of the difficulties in constructing the ontology was adapting the flat XML structure into a complex and detailed RDF ontology. The first step in developing the ontology was to deconstruct the flat structure of the XML and create meaningful object properties (predicates in a RDF triple) that could be used to create semantically-rich RDF. Emphasis was placed on developing object properties that related the Agent to the Creative Work, the Creative Work and Agent to a culture, the Agent to an Event and the Creative Work to an Event. The latter two were used because of the way VRA expresses Agent activity dates and Creative Work creation dates. Both elements use the subelement earliestDate and latestDate. When used in conjunction with an XML type “activity” attribute under the Agent element or with the “alteration”, “broadcast”, “bulk”, “commission”, “creation”, “design”, “destruction”, “discovery”, “exhibition”, “inclusive”, “performance”, “publication”, “restoration”, “view” or “other” attributes under the Date element, it was inferred that what was being expressed in the XML was not a simple/flat data property description, but rather an object property description relating either the Agent to a creative activity event or the Creative Work to a creation event (Table 2 in section 4.4.2). Modeling these VRA attributes as object properties rather than as simple data properties allowed for the creation of a rich semantic relationship that is not possible in simple XML. Figure 7 is a screenshot of object properties stored in Protégé.

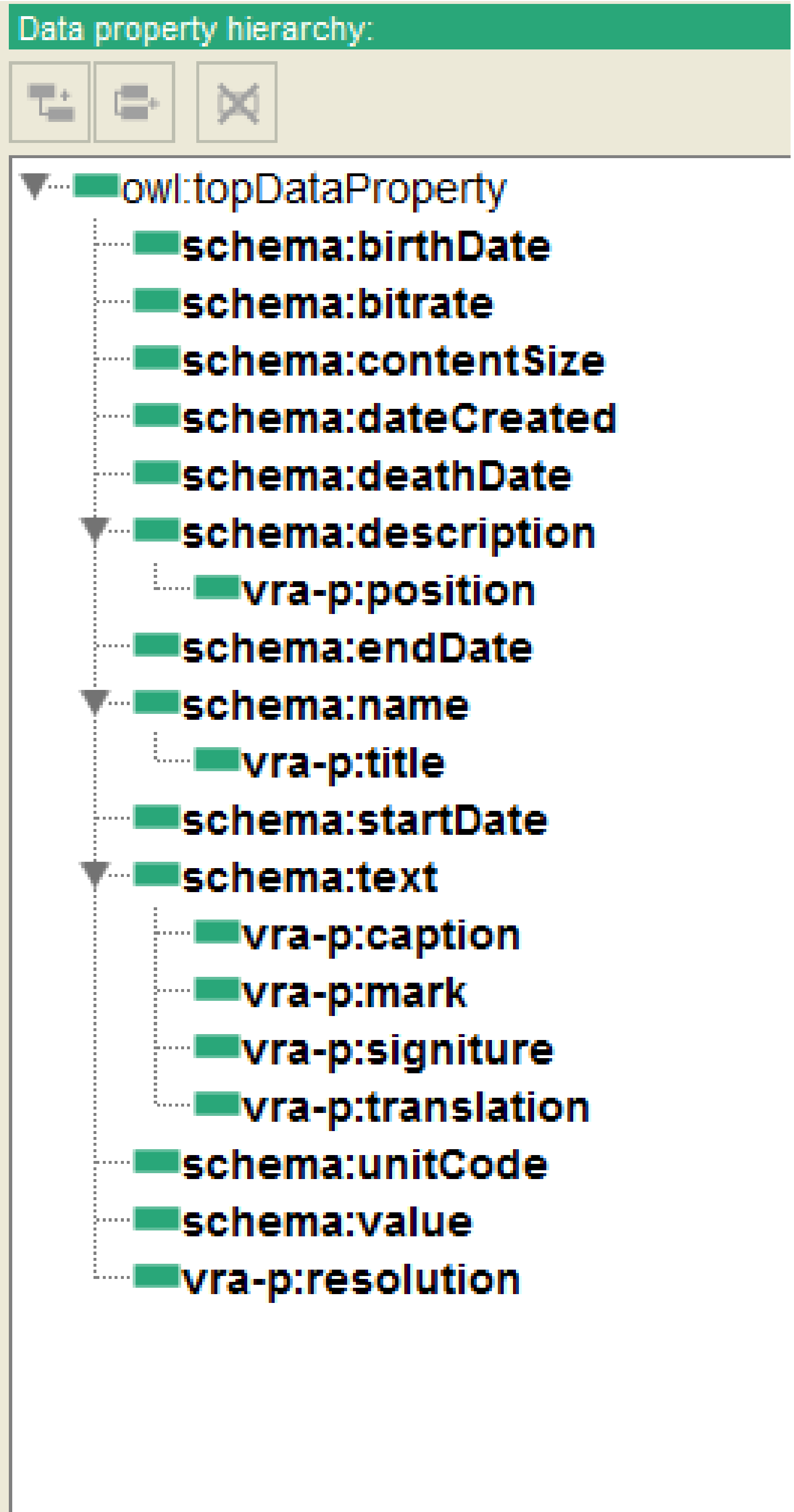


- schema:width
- vra-p:area
- vra-p:base
- vra-p:circumfrence
- vra-p:count ≡ schema:inventoryLevel
- vra-p:creativeActivity
- vra-p:diameter
- vra-p:distanceBetween
- vra-p:duration
- vra-p:hasCulture
- vra-p:hasInscription
- vra-p:hasStylePeriod
- vra-p:hasTechnique
- vra-p:length
- vra-p:locationOf
  - vra-p:formerOwner
  - vra-p:formerRepository
  - vra-p:formerSite
  - vra-p:placeOfCreation
  - vra-p:placeOfDiscovery
  - vra-p:placeOfExhibition
  - vra-p:placeOfInstallation
  - vra-p:placeOfIntention
  - vra-p:placeOfOwnership
  - vra-p:placeOfPerformance
  - vra-p:placeOfPublication
  - vra-p:placeOfRepository
  - vra-p:placeOfSite
- vra-p:material
- vra-p:runningTime
- vra-p:vra:culturalContext
- vra-p:wasAlteration
- vra-p:wasBroadcast
- vra-p:wasCommission
- vra-p:wasCreated
- vra-p:wasDesigned
- vra-p:wasDestroyed
- vra-p:wasDiscovered
- vra-p:wasPerformed
- vra-p:wasPresented
- vra-p:wasProduced
- vra-p:wasPublished
- vra-p:wasRestored

Figure 7. Sample of object properties in the VRA ontology

### Data properties.

Mapping and converting the data properties into the ontology was much less complex than creating the custom object properties. Nevertheless, there were some difficulties in mapping some of the unique VRA attributes into Schema.org. As mentioned earlier, an attempt also was made to include a broader Schema.org data property whenever a unique VRA data property was created. In one instance this required some creative use of both data properties and object properties. Included in some VRA restricted schema is an element called “Inscription”, which is used to describe any inscriptions (such as signatures, dates, captions, etc.) that are on the Work or Image being described. In order to adapt this XML element to RDF, the inscriptions were treated as separate Creative Works that were connected to the Work or Image being described via an object property. Since Schema.org does not have data properties specific enough to detail a signature or a caption, the Schema.org data property “text” was used as a broader property for the VRA “caption”, “mark”, “signature” and “translations” attributes and the Schema.org property “description” was used as a broader property for the VRA “position” attribute (Figure 8).



Data property hierarchy:

- owl:topDataProperty
  - schema:birthDate
  - schema:bitrate
  - schema:contentSize
  - schema:dateCreated
  - schema:deathDate
  - schema:description
    - vra-p:position
  - schema:endDate
  - schema:name
    - vra-p:title
  - schema:startDate
  - schema:text
    - vra-p:caption
    - vra-p:mark
    - vra-p:signiture
    - vra-p:translation
  - schema:unitCode
  - schema:value
  - vra-p:resolution

Figure 8. Data properties in the VRA ontology

### Use of custom elements.

The one area of the resulting ontology that relied heavily on customization was the relationship properties (Protégé object properties). This was due to the fact that almost all of the created object properties were adapted from existing VRA XML attributes and therefore were representative of a highly specific type of VRA element type. For example, although Schema.org has a Schema:contentLocation object property, the VRA model includes many more specific type of attributes that can be connected to the VRA "location" element. In order adequately to address this situation, it was necessary to create custom VRA object properties that specifically addressed all of the individual VRA attributes. Although a few of the VRA attributes could be considered sub-properties of Schema:contentLocation, most required their own unique object properties.

## XSLT Stylesheet

### Overview.

The XSLT Stylesheet (see Appendix A for sample) that was created based on the Notre Dame sample set proved successful both in converting the existing XML into RDF and in parsing the original XML data to form meaningful, machine-readable data. For convenience and to minimize the time needed to find and eliminate mistakes/errors in the XSLT templates, a sub-sample of three records was used in the development of the stylesheet. Since the stylesheet was adapted and created specifically for the sample set, it would be difficult simply to apply it to another data set and expect the same results. However all attempts were made to create enough standard XSLT templates that the stylesheet could be successfully applied to another data set with only minor changes. Even though the stylesheet was designed using the three record sub-sample, the final testing process used the entire 4,150 records in the Notre Dame sample set. The entire XSLT stylesheet is available for download at http://purl.org/jmixter/thesis/.

### Converting XML to RDF.

The resulting XSLT stylesheet was designed as a series of templates that identified and then converted the original XML data into RDF data. Below is a table that illustrates what VRA XML elements were converted RDF.

Table 2: Mappings of VRA XML elements/attributes to RDF and the degrees of matching between the original XML and RDF

| Original XML Element | Original XML Attribute | RDF Output | Matching Degree |
|---|---|---|---|
| vra:collection | | void:DataSet | Close |
| vra:work | | schema:CreativeWork | Broader |
| vra:image | | schema:CreativeWork | Broader |
| vra:agent (name) | personal | schema:Person | Exact |
| vra:agent (name) | corporate | schema:Organization | Exact |
| vra:agent (name) | family | schema:Person | Broader |
| vra:agent (dates) | activity | schema:Event | Close |
| vra:agent (dates) | life | schema:birthdate, schema:deathDate | Exact |
| vra:culturalContext | | vra-p:Culture | Exact |
| vra:date | alteration | schema:Event | Close |
| vra:date | broadcast | schema:Event | Close |
| vra:date | commission | schema:Event | Close |
| vra:date | creation | schema:Event | Close |
| vra:date | design | schema:Event | Close |
| vra:date | destruction | schema:Event | Close |
| vra:date | discovery | schema:Event | Close |
| vra:date | exhibition | schema:Event | Close |
| vra:date | inclusion | schema:Event | Close |
| vra:date | performance | schema:Event | Close |
| vra:date | publication | schema:Event | Close |
| vra:date | restoration | schema:Event | Close |
| vra:title | | schema:name | Exact |
| vra:location | repository | schema:Place | Broader |
| vra:location | publication | schema:Place | Broader |
| vra:location | formerOwner | schema:Place | Broader |
| vra:location | discovery | schema:Place | Broader |
| vra:location | exhibition | schema:Place | Broader |
| vra:location | formerRespository | schema:Place | Broader |
| vra:location | formerSite | schema:Place | Broader |
| vra:location | installation | schema:Place | Broader |

| | | | |
|---|---|---|---|
| vra:location | intended | schema:Place | Broader |
| vra:location | owner | schema:Place | Broader |
| vra:location | performance | schema:Place | Broader |
| vra:location | site | schema:Place | Broader |
| vra:location (name) | corporate | schema:CivicStructure | Exact |
| vra:location (name) | geographic | schema:AdministrativeArea | Broader |
| vra:location (name) | personal | schema:Person | Exact |
| vra:material | | schema:Intangible | Broader |
| vra:stylePeriod | | schema:Intangible | Broader |
| vra:subject | | schema:about | Exact |
| vra:technique | | schema:Intangible | Broader |
| vra:measurements | area | vra-p:area | Exact |
| vra:measurements | base | ra-p:base | Exact |
| vra:measurements | bitDepth | schema:bitrate | Exact |
| vra:measurements | circumference | vra-p:circumference | Exact |
| vra:measurements | count | schema:inventoryLevel | Exact |
| vra:measurements | Depth | schema:depth | Exact |
| vra:measurements | diameter | vra-p:diameter | Exact |
| vra:measurements | distanceBetween | vra-p:distanceBetween | Exact |
| vra:measurements | duration | vra-p:duration | Exact |
| vra:measurements | fileSize | schema:contentSize | Exact |
| vra:measurements | height | schema:height | Exact |
| vra:measurements | length | schema:length | Exact |
| vra:measurements | resolution | vra-p:resolution | Exact |
| vra:measurements | runningTime | vra-p:runningTime | Exact |
| vra:measurements | weight | schema:weight | Exact |
| vra:measurements | width | schema:width | Exact |
| vra:rights | | dcterms:rights | Exact |
| vra:inscription | | schema:CreativeWork | Broader |
| vra:inscription | signature | schema:text | Broader |
| vra:inscription | mark | schema:text | Broader |
| vra:inscription | caption | schema:text | Broader |
| vra:inscription | date | schema:dateCreated | Exact |
| vra:inscription | text | schema:text | Broader |
| vra:inscription | translation | schema:text | Broader |
| vra:relation | relatedTo | schema:isRelatedTo | Exact |
| vra:relation | cartoonFor | vra-p:catroonOf | Exact |
| vra:relation | cartoonIs | vra-p:hasCartoon | Exact |
| vra:relation | componentOf | vra-p:componentOf | Exact |
| vra:relation | componentIs | vra-p:hasComponent | Exact |
| vra:relation | copyAfter | vra-p:copyOf | Exact |
| vra:relation | copyIs | vra-p:hasCopy | Exact |
| vra:relation | counterProofFor | vra-p:counterProofFor | Exact |
| vra:relation | counterProofIs | vra-p:hasCoutnerProof | Exact |
| vra:relation | depicts | vra-p:depicts | Exact |
| vra:relation | depictedIn | vra-p:depictedIn | Exact |
| vra:relation | derivedFrom | vra-p:derivedFrom | Exact |
| vra:relation | sourceFor | vra-p:sourceFor | Exact |
| vra:relation | designedFor | vra-p:designedFor | Exact |

| | | | |
|---|---|---|---|
| vra:relation | contextIs | vra-p:hasContext | Exact |
| vra:relation | exhibitedAt | vra-p:exhibitedAt | Exact |
| vra:relation | venueFor | vra-p:venueFor | Exact |
| vra:relation | facsimileOf | vra-p:facsimileOf | Exact |
| vra:relation | facsimileIs | vra-p:hasFacsimile | Exact |
| vra:relation | formerlyPartOf | vra-p:fomerlyPartOf | Exact |
| vra:relation | formerlyLargerContextFor | vra-p:formerlyLargerContextFor | Exact |
| vra:relation | imageOf | vra-p:imageOf | Exact |
| vra:relation | imageIs | vra-p:hasImage | Exact |
| vra:relation | mateOf | vra-p:mateOf | Exact |
| vra:relation | mateIs | vra-p:hasMate | Exact |
| vra:relation | modelFor | vra-p:modelFor | Exact |
| vra:relation | modelIs | vra-p:hasModel | Exact |
| vra:relation | partOf | vra-p:part of | Exact |
| vra:relation | largerContextFor | vra-p:largerContextFor | Exact |
| vra:relation | partnerInSetWith | vra-p:partnerInSetWith | Exact |
| vra:relation | pendentOf | vra-p:pendentOf | Exact |
| vra:relation | planFor | vra-p:planFor | Exact |
| vra:relation | planIs | vra-p:hasPlan | Exact |
| vra:relation | preparatoryFor | vra-p:preparatoryFor | Exact |
| vra:relation | basedOn | vra-p:basedOn | Exact |
| vra:relation | printingPlateFor | vra-p:printingPlateFor | Exact |
| vra:relation | printingPlateIs | vra-p:hasPrintingPlate | Exact |
| vra:relation | prototypeFor | vra-p:prototypeFor | Exact |
| vra:relation | prototypeIs | vra-p:hasPrototype | Exact |
| vra:relation | reliefFor | vra-p:feliefFor | Exact |
| vra:relation | reliefIs | vra-p:hasRelief | Exact |
| vra:relation | replicaOf | vra-p:replicaOf | Exact |
| vra:relation | replicaIs | vra-p:hasReplica | Exact |
| vra:relation | studyFor | vra-p:studyFor | Exact |
| vra:relation | studyIs | vra-p:hasStudy | Exact |
| vra:relation | versionOf | vra-p:versionOf | Exact |
| vra:relation | VersionIs | vra-p:hasVersion | Exact |
| vra:workType | | -- | |

One of the difficulties in mapping VRA 4.0 into RDF was determining what type of relationship would be created between the original XML tags and the newly created RDF classes. Additionally, not all of the original VRA elements converted into corresponding Schema.org classes. The vra:measurements, vra:relation and vra:subject elements all were converted into Schema.org object properties. The vra:title element was converted directly into a Schema.org data property.

For many of the XML tags, there existed a one-to-many relationship. The vra-date element is an excellent example of this problem. Since the vra-date tag can be used to describe many different types of dates associated with the item, it was necessary to use the XML attributes to help determine what RDF class the tag should be converted into. The vra-date element also was unique, in that in many cases the resulting RDF class was schema:Event. The vra:date type attribute “broadcast” serves as a great example of this conversion. In the original XML data, this element attribute combination is not used simply to describe a date. Rather, it is used to describe a time-span (i.e. event) during which the item being described (for example an audio recording) was broadcast. In order accurately to translate the semantics of this element attribute combination into RDF, it was necessary to map it to a schema:Event class. The name of the attribute (for example broadcast) was used as the name of the schema:Event, and the date values (vra:earliestDate and vra:latestDate) were used to create scheme:startDate and schema:endDate properties to describe when the schema:Event occurred.

All of the broader SKOS relationships represent the attempt to map the existing VRA element to a popular domain specific vocabulary. In all of these cases, the original VRA elements were retained and converted into RDF in order to preserve the specific meaning of the vra:relation element. Table 2 only shows the broader match in order to highlight how extremely detailed models such as VRA 4.0 can successfully incorporate popular domain specific vocabularies. The one exception to this is the vra-relation element. Since this element contains extremely specific attributes that in some cases radically change the meaning of “relation”, the exact match was used in Table 2. For these elements, schema:isRelatedTo was used as a supra-property and was included in the RDF output in order to create a valuable (albeit a very high-level) semantic relationship.

The vra:workType was the one example of a one-to-none relationship and consequently it did not have a resulting RDF class or property. This was because when the element value was converted into a direct triple, it became a simple description of what the item was. For example, an image that had a vra:workType of digital photograph would be converted into an RDF statement saying that the image is a digital photograph.

**Additional domain specific vocabularies.**

The vra:collection, vra:agent and vra:rights elements required special data models to accommodate the type of data that they included. Since the vra:collection description from the Notre Dame sample set was not actually an item that could be described using RDF, it was difficult to find an appropriate way to describe accurately the concept of a collection. In order to describe the rather abstract concept of a collection, the VoID data model was selected. While schema:Person and schema:Organization were adequate for translating the vra:agent (type:personal and type:family) and vra:agent (type:corporate), there were no Schema.org classes that adequately covered the entire scope of vra:agent. Consequently the FOAF vocabulary was used since it has a class (FOAF:Agent) that directly matched the vra:agent.

Vra:rights was another element that was rather difficult to match with a Schema.org class. Schema.org does have a schema:publishingPrinciples data property, but it requires the use of a URL to point to the specific rights associated with the item. Additionally, the definition of the data property was too vague to associate directly with the vra:rights element. As an alternative, DC Terms (http://dublincore.org/documents/dcmi-terms/) was chosen as an appropriate data model link with vra:rights. As with the schema:publishingPrinciples data property, the dcterms:rights data property has a set range of a URL. In order to overcome this issue, the XSLT stylesheet was designed to convert vra:rights into dcterms:rights, but then leave the nod blank (i.e. not fill in a URL linking to the specific rights statement). Instead of using a URL, the stylesheet created a rdfs:label data property within the cdterms:rights property and then extracted the string value used in the original XML data. Although this was not the ideal way to express rights in RDF, it was the best method that accurately and effectively converted the existing concepts and values without losing any data. Below is an example of how this unique conversion process worked:

```
<dcterms:rights>
        <rdf:Description>
                <rdfs:label>scans retain Creative Commons license</rdfs:label>
        </rdf:Description>
</dcterms:rights>
```

**Conversion of VRA XML attributes.**

Some of the VRA elements listed in Table 2 were divided into multiple templates that matched on specific VRA attributes. These unique templates were used to parse out specific VRA attribute values in order to create a more comprehensive RDF output. This process helped insure that both high level Schema.org elements as well as specific custom VRA elements were included in the RDF output. For example, within the vra:location element, there were 12 unique templates that were used to identify specifically the type of location and how it was related to the item being described.

### Creation of URIs.

#### *Controlled vocabularies.*

One of the most important tasks during the development of the XSLT stylesheet was to enhance the original XML data and produce RDF data that a machine could read and understand. One way that this was successfully implemented in the final stylesheet was through the creation of URIs in the RDF output. Since the original XML data already used controlled vocabularies with reference IDs, the stylesheet was designed to parse out the relevant data and then recombine it to form a unique URI in the RDF output:

```
<xsl:when test="@vocab ='LCSAF'">
        <vra-p:hasStylePeriod>
        <rdf:Description>
                <xsl:attribute name="rdf:about">
                        <xsl:text>http://id.loc.gov/authorities/subjects/</xsl:text>
                        <xsl:value-of select="translate(@refid,' ','')"/>
                </xsl:attribute>
                <rdf:type rdf:resource="http://schema.org/Intangible"/>
                <rdf:type rdf:resource="http://purl.org/vra/StylePeriod"/>
                <schema:name><xsl:value-of select="."/></schema:name>
        </rdf:Description>
        </vra-p:hasStylePeriod>
</xsl:when>
```

The example above illustrates how the stylesheet was able to identify LCSAF headings, parse out the relevant information and recombine it to form an information-rich URI. In addition to generating URIs for established Linked Data vocabularies, the stylesheet was also designed to create URIs out of vocabularies that are not currently published as Linked Data. In particular, the study focused on Getty vocabularies, which are frequently used in cataloging visual objects. In order to demonstrate how the various Getty vocabularies successfully could be published as Linked Data, a standard format was developed and then templates were created that could identify Getty vocabularies and parse out the important information from the XML data:

```
<xsl:when test="@vocab ='AAT'">
        <vra-p:hasStylePeriod>
        <rdf:Description>
                <xsl:attribute name="rdf:about">
                        <xsl:text>http://purl.org/getty/vocab/</xsl:text>
                        <xsl:value-of select="@vocab"/>
                        <xsl:text>/</xsl:text>
                        <xsl:value-of select="@refid"/>
                </xsl:attribute>
                <rdf:type rdf:resource="http://schema.org/Intangible" />
                <rdf:type rdf:resource="http://purl.org/vra/StylePeriod" />
                <schema:name><xsl:value-of select="."/></schema:name>
        </rdf:Description>
        </vra-p:hasStylePeriod>
</xsl:when>
```

The example above illustrates how the stylesheet was designed to identify when an AAT heading was used and then extract the valuable data to form a proposed URI.

### *Parsing XML elements.*

To parse successfully the various types of controlled vocabulary headings in the sample set, unique sub-templates were created that matched on specific XML “vocab” attributes. The XSLT sample above illustrates a sub-template of the vra:stylePeriod template. The template is set to test whether the XML “vocab” attribute is AAT. If it the statement is true, then the stylesheet runs the sub-template. If it is false, then the sub-template is skipped. In order to make the XSLT stylesheet as universal as possible, sub-templates for LCSH, LCSAF, LCNAF, LCTGM, AAT, ULAN and TGN were used in all of the templates that had corresponding controlled vocabulary terms. Since the VRA core documentation only provides suggested vocabularies for controlled heading use, including those seven unique vocabulary sub-templates was seen as an efficient way to control for catalogers that deviated from the VRA 4.0 restricted schema.

### Eliminating unnecessary data.

The final step in the XSLT stylesheet was to identify data that was not converted into RDF and comment it out so it did not cause problems during validation. The two templates below were used to successfully identify left over data. Once that data was found, it was commented out and marked with the phrase “skipping over”.

```
<xsl:template match="*">
        <xsl:comment>
                <xsl:text>Skipping over </xsl:text>
                <xsl:value-of select="name()"/>
        </xsl:comment>
        <xsl:apply-templates />
</xsl:template>

<xsl:template match="@*|text()">
        <xsl:comment>
                <xsl:text>Skipping over </xsl:text>
                <xsl:value-of select="."/>
        </xsl:comment>
</xsl:template>
```

Since the templates only mark and comment out the data that is skipped over by the rest of the stylesheet, it still is possible to review the initial RDF output in order to make sure that no important or valuable data was skipped over. Additionally, since the template comments out the skipped data, there is no need to manually go back and delete data from the final RDF/XML output.

## RDF Output Data

### Validation.

The resulting RDF output (see Appendix E for comparison between original XML data and RDF/XML output data) was run through the W3C RDF Validator in order to make sure that the mechanics of the stylesheet worked properly. All of the validated RDF output (i.e. all 4,150 XML records converted into RDF descriptions) used the W3C service. This was an important first step, in that it proved that the XSLT stylesheet was successful in converting XML into RDF.

### Manual review.

Additionally, the output was manually reviewed in order to make sure that all of the templates worked correctly and produced meaningful and accurate RDF (see Appendix F for sample RDF output in Turtle). This second step was very important because the W3C RDF Validator only checked for mechanical correctness. There is no way, other than manual review, to confirm that the RDF that is being produced is of a high quality. Although only three VRA records were used to build and test the XSLT stylesheet, the entire Notre Dame sample (4,150 records) was tested for the final part of the study. The RDF output described 5,382 unique entities. The output included 40,542 described entities in total. In reviewing the RDF output, all of the templates created in the original XSLT stylesheet work correctly and created correct and accurate RDF/XML. It took 24 working hours to manually review the final RDF/XML for accuracy and completeness.

#### *Reviewing templates.*

One of the first tasks in reviewing the RDF output was to confirm that all of the XSLT templates were called correctly and that they converted the original XML data properly. Since all of the mechanical problems of the stylesheet were corrected using the small sample of three records, the process of reviewing the entire Notre Dame sample focused primarily on ensuring that the individual templates produced correct RDF for each of the records. Each of Collection, Work and Image descriptions was reviewed individually for accuracy and completeness. During the review, a checklist of all of the appropriate XSLT templates was referenced to make sure that they were implemented correctly in creating the RDF output. Additionally, all of the commented out data was reviewed to make sure that no valuable data was lost in the conversion process. In order to confirm that the RDF being generated by the stylesheet was of high quality, the output of the XSLT sub-templates, which were used for converting controlled vocabulary headings into URIs, were reviewed for accuracy. Below is a comparison of the original XML data with the resulting RDF/XML output once it was run through the XSLT stylesheet:

```
<subjectSet>
        <display>architecture; rulers and leaders; saints; Julius II, Pope, 1443-1513; Peter, the
Apostle, Saint</display>
        <notes />
        <subject>
                <term type="personalName" vocab="LCNAF" refid="n 80030746">Julius II,
Pope, 1443-1513</term>
        </subject>
        <subject>
                <term type="personalName" vocab="LCNAF" refid="n 79022118">Peter, the
Apostle, Saint</term>
        </subject>
</subjectSet>
```

```
<schema:about>
        <rdf:Description rdf:about="http://id.loc.gov/authorities/names/n80030746">
                <rdf:type rdf:resource="http://schema.org/Person"/>
                <schema:name>Julius II, Pope, 1443-1513</schema:name>
        </rdf:Description>
</schema:about>
<schema:about>
        <rdf:Description rdf:about="http://id.loc.gov/authorities/names/n79022118">
                <rdf:type rdf:resource="http://schema.org/Person"/>
                <schema:name>Peter, the Apostle, Saint</schema:name>
```

```
</rdf:Description>
</schema:about>
```

The "rdf:about" value is the resulting URI that was created when the original XML data was run through the XSLT stylesheet. All of the XSLT sub-templates that were responsible for creating either working URIs or proposed URIs (for the Getty vocabularies) functioned successfully in converting the flat XML data in semantically-rich RDF.

### *Problems in RDF output.*

Although a few problems were encountered during the manual review of the RDF output, they all were related to the original XML data. In a few instances, the cataloger of the Notre Dame sample set used the incorrect vocab attribute:

```
<term type="personalName" vocab="LCSAF" refid="n 84216944">Molteni, A.</term>
```

The example above illustrates an instance in which the original XML data incorrectly identified this term as a LCSAF heading, when it actually is a LCNAF heading. During the conversion process, the XLST stylesheet matched on the vocab attribute LCSAF and consequently ran the LCSAF sub-template. As a result the resulting RDF URI was incorrect:

```
<schema:about>
        <rdf:Description rdf:about="http://id.loc.gov/authorities/subjects/n84216944">
                <rdf:type rdf:resource="http://schema.org/Person" />
                <schema:name>Molteni, A.</schema:name>
        </rdf:Description>
</schema:about>
```

Even though in this instance the URI was incorrectly converted, measures were taken to mitigate the problem of incorrect sample data. In addition to matching on the "vocab" attribute, all of the vocabulary sub-templates were also designed to use the "type" attribute to identify the correct

Schema.org class to associate with the description. Even though the converted URI was incorrect, the RDF output was able to match on the “type” attribute (in this case “personalName”) and accurately identify the entity as a schema:Person. It was not part of this study to use the XSLT stylesheet to retroactively correct original cataloging data, and therefore these few mistakes were ignored.

# Chapter V

# Discussion

## Overview

The study successfully demonstrated that existing data models can be updated with new Linked Data vocabularies and that existing data can be updated and published as Linked Data. The following sections include recommendations for how the results of this study can be used by organizations, limitations that affected the research and a comprehensive conclusion detailing the major aspects of the study. Finally, this chapter identifies ideas for future projects that could build upon the findings of this study as well as further examine and test the value of publishing Linked Data.

## Recommendations

The results of this study demonstrated a potential approach and applicable tool that can be used by institutions that have used VRA Core 4.0 in converting their existing metadata XML records into RDF triples. Judging by the on information provided by the major search engines, doing so would increase data visibility on the Internet. Nevertheless, it should be noted that the availability of Linked Data vocabularies would have a direct impact on such an effort. While the resulting RDF output data contained valid URIs for all of the Library of Congress controlled headings, the URIs for the Getty vocabularies, which are predominantly used for cataloging visual material, were only proposed and did not resolve to online descriptions. It is hoped that the proposed URIs, as well as the ontology developed for this study, could be utilized by the Getty Research Institute to model its existing vocabularies for publication as Linked Data. Any adoption of the results derived from this study should, by the time of implementation, consult the available Linked Data vocabularies to be used, especially the ones published by the Getty Research Institute.

In order for the XSLT stylesheet to be implemented effectively by an institution, it is important that their original data is of high quality and conforms as closely as possible to the VRA 4 restricted schema. To avoid making changes to the XSLT stylesheet, it is recommended that organizations that currently are using and creating VRA data, also use the schema's proposed element and attribute names. This will help facilitate an easier implementation of the existing stylesheet and prevent the need to make any changes in order for it to work with other datasets.

If existing data does use different values for element or attribute names, organizations can make changes in the stylesheet to address those variances. The primary changes would include replacing the existing template "match" value from the VRA 4 restricted schema name to that used locally. In the same fashion, the attribute values that differ from the VRA restricted schema also would have to be changed in the stylesheet. Finally, if any additional vocabularies are used, additional sub-templates would have to be created. If the vocabulary already is published as Linked Data, then the Library of Congress sub-templates can be used for reference and as examples. If the vocabulary is not published as Linked Data, then the Getty sub-templates will need to be used for reference. This method also can be used to illustrate/model how local controlled vocabularies could be generated and represented as URIs for Linked Data publication.

Although this study used a “clean” record collection as a sample set, it is recognized that there is an abundance and proliferation of “dirty data”, and that this “dirty data” might not work well with the XSLT stylesheet that was developed for this study. In order to address this issue, it is recommended that organizations conduct a data review, similar to the one conducted for this study, of any datasets that might be used with the XSLT stylesheet. If problems or issues are found during the review, it is advised that a comprehensive analysis and normalization project be conducted before the stylesheet is run against the data. If it is found that the problems are minor, then modifications to the stylesheet can be made that account for the issues and automatically normalize the RDF output. An example of a minor issue would be to modify the stylesheet to identify three different string formats for identifying a work type (i.e. PDF, .pdf and pdf), and then to normalize them all to one standard style.

With regards to the stylesheet created during this study, it is recommended that any future researchers pay close attention to its original purpose. The stylesheet was not developed with the intention of being used by future catalogers in their routine cataloging practices. Rather, it was developed to prove that existing data could be updated to meet the new data standards. Additionally, the stylesheet was designed to allow organizations to convert their existing legacy data into modern RDF triples for future use in a RDF database (i.e. a triplestore). For current and future catalogers, the ontology that was developed for this study is much more important than the actual stylesheet itself and should be used to help rethink the way in which catalogers describe items. It should be understood that data conversion, regardless of the method used, is a “lossy” process. Therefore, it is strongly advised that the results of this study be used as motivation/evidence for the reevaluation of cataloging practices, rather than being used simply to convert metadata in order to preserve legacy cataloging practices, which are in need of updating.

## Limitations of the Study

### Specificity of the stylesheet.

This study was limited by the need for the XSLT stylesheet to be designed specifically for the Notre Dame sample set. Since VRA 4.0 is a recommended set of elements and attributes, there is no requirement that institutions follow a specific template. This allows institutions to change the names of elements or attributes, and also to include as much or as little detail about each element as they see fit. Consequently, this stylesheet, while applicable to institutions that follow the VRA 4.0 template, was designed specifically for the sample data and therefore could not directly be applied to other sample datasets. In order for the stylesheet to work with other datasets, it will require minor modifications. A benefit of using XSL is that, if an institution did follow the VRA 4.0 restricted schema but supplemented it with additional material, the XSL stylesheet still will be able to pull out the valid VRA elements and attributes and simply skip over the data that is not applicable.

### Use of VRA restricted schema.

In addition to the specificity of the XSLT stylesheet, this study also was limited by the fact that the VRA ontology and the stylesheet both were based on the VRA restricted schema. No attempt was made to incorporate elements or attributes from the unrestricted schema, and consequently the results of the study are limited to use by organizations that follow the VRA restricted schema.

### File and data formats.

Since the XSLT stylesheet was created to convert XML documents into RDF/XML, the use of the stylesheet consequently is reliant on the use of XML as a starting file format. Although it is possible to convert CSV files and relational database entries into XML format, it is advisable that any such conversion process be carefully reviewed for accuracy and consistency.

Since data conversion is inherently “lossy”, a two-step conversion process should be avoided if possible.

In addition to the file format, the format of the raw data also was a limitation in this study. Since the stylesheet was based on the sample set, the templates were designed to match on specific XML element and attribute values used throughout the dataset. The sample dataset also was used to determine which templates searched for controlled vocabulary terms, and also which vocabularies were searched for (i.e. LCSH, ULAN or AAT).

## Conclusion

The problem addressed in this study was how to develop and incorporate an automatic method for converting VRA Core 4.0 XML records into RDF/XML records. To meet the primary objective of creating an effective and efficient way to convert existing VRA 4.0 records into Linked Open Data, the study attempted to develop a methodology that could be used to accurately and affectively convert VRA Core 4.0 records into rich semantic RDF Linked Data, as well as develop a VRA ontology that uses classes and properties from popular domain specific vocabularies. It also met another strategic objective, namely to create an XSLT stylesheet that would enable current users of VRA Core 4.0 to convert records into rich RDF/XML style records. The major motivation of the study was to determine how effectively legacy metadata could be converted into RDF, and subsequently added to the ever-growing Semantic Web. Additionally, the study attempted to create a data model that was, in essence, an aggregation of existing data models.

There are many reasons to publish VRA data as Linked Data and to do so in a way that leverages popular domain specific vocabularies. First of all, it is clear that over the past decade Linked Data has become a popular means both of publishing data as well as sharing data across the Internet.  The BBC, the New York Times, dbpedia.org and Freebase.com all are well-known leaders in creating and using Linked Data. In 2012, OCLC announced that it would be implementing Linked Data into its WorldCat.org search service, as well as publishing FAST as a Linked Data vocabulary.  In addition to OCLC's FAST vocabulary, the Library of Congress also has published its vocabularies as Linked Data. Many bibliographic datasets have been produced by libraries and other memory institutions during the past five years; however, no VRA record-based dataset has been reported as of today (April 21, 2013).

VRA is a widely used data model standard for cataloging visual resources. The VRA 4.0 data model was approved by the METS Editorial Board in 2007, and currently hosted by the Library of Congress. A 2011 study of 103 voluntary respondents showed that 84 of those collections are using the VRA data model (Carter, Double, Rose-Sander, & Webster, 2011).  Of those, 56 were using VRA Core 4.0. VRA Core has not been released as a Linked Data-enabled model. A similar model called CIDOC, which also was designed for describing cultural items and recently was published as a Linked Data model, was built from the ground up using entirely custom RDF classes and properties (instead of using RDF classes and properties from popular domain specific vocabularies). The practice of using custom classes and properties to develop a data model contradicts one of the W3C's core suggestions about Linked Data: namely to use industry standards as much as possible. Given the limitations of both of these data models, this study was conducted with the purpose of maximizing interoperability through using popular domain specific vocabularies to the greatest extent possible.  Schema.org, a model/ontology published by the major search engines (Google, Yahoo, Bing and Yandex) in 2012 for describing things in RDF, was chosen as the major backbone of the study. Although the Schema.org model takes a relatively high-level approach in defining classes and properties, it still is detailed enough to cover a wide variety of disciplines and industries. A relevant example is the WorldCat Linked Data records that were produced using schema.org model. An added benefit of Schema.org is that data which is described using the model is understood and indexed by the major search engines.

The study developed an ontology to explain the new data model and an XSLT stylesheet that could be used to convert the VRA XML into RDF.  Sample records from the VRA website, as well as a sample collection (the Notre Dame Lantern Slide collection) provided by a member of the VRA listserv, were used as the basis for both the VRA Core Ontology and the corresponding XSLT stylesheet. The flat hierarchical structure of XML schemas made the development of the ontology rather complex.  In order to convert the VRA schema into a meaningful ontology, the XML elements had to be picked apart and the associated attributes used to build object properties in the ontology. Together, the VRA Core Ontology consisted of 34 ontological classes and a total of 169 properties. Thirty-five were expressed with classes or properties from schema.org, with additional ones coming from FOAF (1), DCMI Metadata Terms (1), while 127 were proposed as custom VRA elements. Of the 127 custom VRA elements, 84 were structured as sub-classes or sub-properties of established Schema.org elements.  Since the resulting RDF/XML output includes both sub-class/sub-property as well as super-class/super-property descriptions, 66% of the custom VRA classes/properties still will be understood by any machine that understands the Schema.org data model.  The resulting ontology was published and made available on a PURL website.  This was to ensure that URL remains consistent regardless of where the ontology is hosted.

The XSLT stylesheet was instrumental in implementing the data model and producing RDF triples from VRA XML records. As the XSLT stylesheet was being developed, it continually was tested using three sample records from the Notre Dame sample set. This creation-test-revise process was used in order to ensure that the XSLT templates worked properly and that revisions to the stylesheet did not result in any retroactive problems with templates that were previously working. In order to add value to the resulting RDF output, a set of 50 XSLT templates was designed to parse out data from the original XML in order to form URIs. Since the Getty vocabularies are not yet published as Linked Data, proposed URIs were generated from any Getty controlled headings used in the sample set. Once the stylesheet was developed and tested using the three sample records from the Notre Dame sample set, the stylesheet then was run against the entire Notre Dame sample set (4,150 records).

The XSLT stylesheet successfully converted all 4,150 Notre Dame sample records into RDF/XML. The mechanical structure of the RDF was confirmed using the W3C RDF Validator, and then the RDF output was reviewed manually to check for problems resulting from errors in the XSLT templates. In addition to converting the original XML elements and attributes into RDF classes and properties, the stylesheet also was able to parse out data related to controlled vocabulary headings and form URIs. Valid URIs were constructed for all of the Library of Congress controlled headings. Getty vocabulary terms were converted into proposed URIs, with the intention that they might be used to illustrate how the Getty organization could convert its existing vocabularies into Linked Data.

## Future Studies

### Search Engine Optimization.

The results of this study have established the groundwork for future studies to examine further the empirical value of publishing Linked Open Data. This research could be conducted through the examination of end-user tools and resources which can be developed through the use of aggregated data from the Semantic Web. Tools such as Google’s Knowledge Graph also could be studied and tested to determine if improving existing datasets is possible through mass aggregation of information across the Semantic Web. For libraries, museums and archives, studies could be conducted to determine if end-user tools/resources improve patron satisfaction and information discovery. The value of publishing Linked Open Data also could be evaluated by studying whether data published in RDF formats leads to improved visibility by search engines. For libraries, museums and archives, this could help determine whether Linked Data could be used to direct/guide patrons to their websites through search engine results. OCLC's effort to implement Linked Data into WorldCat.org could be used as the basis for analyzing the effectiveness of improving search engine visibility.

A possible future use case for this research would be to study a website for an art gallery. User visits, click-through and search engine rankings could be analyzed to determine the SEO of the original website. After this initial data is collected, the art gallery website could be redesigned using RDFa as opposed to basic HTML. The same data collection (user visits, click-through, and search engine ranking) could be conducted following the redesign of the website. By comparing the differences in the data results, one would be able to provide empirical evidence showing the difference that using Linked Data (in this case RDFa) makes in data visibility and discovery. Currently, the stylesheet is for converting the VRA Core 4 XML records into triples, which might be more detailed than is needed in a Website's microdata. Therefore, there would be a new use case for this research to study and validate, and develop appropriate core elements for the microdata use.

**Development of more Linked Data datasets.**

Since a majority of the vocabulary terms used in the sample set come from vocabularies that are not yet published as Linked Data, there is a possibility that results from the project could be used to help model how these vocabularies could be transitioned into Linked Data. In particular, publishing the Getty TGN and AAT vocabularies would be extremely important for any image catalogers who wish either to catalog using Linked Data or to convert their legacy records into Linked Data. The results of this project have produced viable URI patterns for both the Getty's AAT and TGN vocabularies. Since Getty's ULAN vocabulary has been adopted into VIAF, it was deemed unnecessary and counterproductive to construct hypothetical URI for these terms. With minor adjustments, the ULAN terms could be constructed into Getty-specific URIs, in the event that there was an interest in publishing all of their vocabularies as Linked Data. Another future research task would be turning the value lists (such as roles of artists) into independent datasets, following the example of the Library of Congress MARC Relator vocabulary.

**New cataloging practices for cultural objects and visual resources.**

A major area of future research lies in the evaluation of existing cataloging practices in describing cultural objects and visual resources. The results of this study have provided catalogers both with a tool to convert legacy VRA metadata and with a data model that could serve as a basis for retooling how cultural objects and visual resources are described. As the publishing of Linked Data continues to grow, it will become increasingly important for information organizations to adopt the best practices of creating and sharing data. Currently, these practices include not using proprietary file formats and creating data that easily can be linked with related data (both within and outside of the field of expertise). In order to follow through on these best practices, catalogers will need to address and change their current practices and begin to create descriptive data that can be implemented and consumed by the Semantic Web.

# Appendices

## A. XSLT Stylesheet Sample[2]

```
<!-- Agent template -->
<xsl:template match="vc:agent">
    <schema:creator>
    <rdf:Description>
        <xsl:if test="vc:name/@refid">
            <xsl:attribute name="rdf:about">
                <xsl:text>http://viaf.org/viaf/sourceID/JPG%7C</xsl:text>
                <xsl:value-of select="vc:name/@refid"/>
                <xsl:text>#skos:Concept</xsl:text>
            </xsl:attribute>
        </xsl:if>
        <rdf:type rdf:resource="http://xmlns.org/foaf/0.1/Agent" />
        <xsl:if test="vc:name/@type='personal'">
            <rdf:type rdf:resource="http://schema.org/Person" />
        </xsl:if>
        <xsl:if test="vc:name/@type='corporate'">
            <rdf:type rdf:resource="http://schema.org/Organization" />
        </xsl:if>
        <xsl:choose>
        <xsl:when test="vc:name/@type='family'">
            <schema:familyName>
                <xsl:value-of select="vc:name" />
            </schema:familyName>
        </xsl:when>
        <xsl:otherwise>
            <schema:name>
                <xsl:value-of select="vc:name" />
            </schema:name>
        </xsl:otherwise>
        </xsl:choose>
        <xsl:if test="vc:dates[@type='life']">
            <schema:birthDate>
                <xsl:value-of select="vc:dates/vc:earliestDate"/>
            </schema:birthDate>
            <schema:deathDate>
                <xsl:value-of select="vc:dates/vc:latestDate"/>
            </schema:deathDate>
        </xsl:if>
        <xsl:if test="vc:dates[@type='activity']">
            <vra-p:creativeActivity>
            <rdf:Description>
            <rdf:type rdf:resource="http://schema.org/Event" />
                <schema:name>ArtisticActivity</schema:name>
                <schema:startDate>
                    <xsl:value-of select="vc:dates/vc:earliestDate" />
                </schema:startDate>
                <schema:endDate>
                    <xsl:value-of select="vc:dates/vc:latestDate" />
                </schema:endDate>
            </rdf:Description>
            </vra-p:creativeActivity>
        </xsl:if>
        <xsl:if test="vc:culture">
        <vra-p:hasCulture>
        <rdf:Description>
```

[2] The prefix vc: was used throughout the stylesheet as opposed to vra:

```
                        <xsl:attribute name="rdf:about">
                                <xsl:text>http://purl.org/getty/vocab/ulan/</xsl:text>
                                <xsl:value-of select="vc:culture"/>
                        </xsl:attribute>
                        <schema:name>
                                <xsl:value-of select="vc:culture" />
                        </schema:name>
                </rdf:Description>
                </vra-p:hasCulture>
                </xsl:if>
        </rdf:Description>
        </schema:creator>
</xsl:template>
```

## B. Generated URI for Library of Congress Term

```
<schema:about>
        <rdf:Description rdf:about="http://id.loc.gov/authorities/names/n80030746">
                <rdf:type rdf:resource="http://schema.org/Person" />
                <schema:name>Julius II, Pope, 1443-1513</schema:name>
        </rdf:Description>
</schema:about>
```

## C. Proposed URI for Getty Vocabulary Term

```
<schema:containedIn>
        <rdf:Description rdf:about="http://purl.org/getty/vocab/TGN/7001168">
                <rdf:type rdf:resource="http://schema.org/Place" />
                <schema:url>http://www.getty.edu/vow/TGNFullDisplay?find=7001168&amp;place=&amp;nation=&amp;p
                rev_page=1&amp;english=N&amp;subjectid=7001168
                </schema:url>
                <schema:name>Rome (Vatican City)</schema:name>
        </rdf:Description>
</schema:containedIn>
```

## D. Blank Node RDF Output

```
<vra-p:material>
        <rdf:Description>
                <rdf:type rdf:resource="http://purl.org/vra/Material" />
                <schema:name>marble</schema:name>
        </rdf:Description>
</vra-p:material>
```

## E. Original VRA XML compared to RDF/XML

### Original VRA XML.

```
<agentSet>
        <display>Carlo Maderno (Italian architect, ca. 1556 -1629); Donato Bramante (Italian architect, 1444-1514);
Michelangelo Buonarroti (Italian architect, 1475-1564); Pirro Ligorio (Italian architect, ca.1500-1583) and others</display>
        <notes />
<agent>
        <name vocab="ULAN" refid="500007668" type="personal">Maderno, Carlo</name>
        <dates type="life">
                <earliestDate>1556</earliestDate>
                <latestDate>1629</latestDate>
        </dates>
        <culture>Italian</culture>
        <role>architect</role>
</agent>
agent>
        <name vocab="ULAN" refid="500019098" type="personal">Bramante, Donato</name>
        <dates type="life">
                <earliestDate>1444</earliestDate>
                <latestDate>1514</latestDate>
        </dates>
        <culture>Italian</culture>
        <role>architect</role>
</agent>
<agent>
        <name vocab="ULAN" refid="500018431" type="personal">Ligorio, Pirro</name>
        <dates type="life">
                <earliestDate>1500</earliestDate>
                <latestDate>1583</latestDate>
        </dates>
        <culture>Italian</culture>
        <role>architect</role>
</agent>
<agent>
        <name vocab="ULAN" refid="500010654" type="personal">Buonarroti, Michelangelo</name>
        <dates type="life">
                <earliestDate>1475</earliestDate>
                <latestDate>1564</latestDate>
        </dates>
        <culture>Italian</culture>
        <role>architect</role>
</agent>
</agentSet>
```

### RDF/XML.

```
<schema:creator>
        <rdf:Description rdf:about="http://viaf.org/viaf/sourceID/JPG%7C500007668#skos:Concept">
                <rdf:type rdf:resource="http://xmlns.org/foaf/0.1/Agent" />
                <rdf:type rdf:resource="http://schema.org/Person" />
                <schema:name>Maderno, Carlo</schema:name>
```

```
    <schema:birthDate>1556</schema:birthDate>
    <schema:deathDate>1629</schema:deathDate>
    <vra-p:hasCulture>
        <rdf:Description rdf:about="http://purl.org/getty/vocab/ulan/Italian">
            <schema:name>Italian</schema:name>
        </rdf:Description>
</vra-p:hasCulture>
</rdf:Description>
</schema:creator>
<schema:creator>
    <rdf:Description rdf:about="http://viaf.org/viaf/sourceID/JPG%7C500019098#skos:Concept">
        <rdf:type rdf:resource="http://xmlns.org/foaf/0.1/Agent" />
        <rdf:type rdf:resource="http://schema.org/Person" />
        <schema:name>Bramante, Donato</schema:name>
        <schema:birthDate>1444</schema:birthDate>
        <schema:deathDate>1514</schema:deathDate>
        <vra-p:hasCulture>
            <rdf:Description rdf:about="http://purl.org/getty/vocab/ulan/Italian">
                <rdf:type rdf:resource="http://purl.org/vra/Culture" />
                <schema:name>Italian</schema:name>
            </rdf:Description>
        </vra-p:hasCulture>
</rdf:Description>
</schema:creator>
<schema:creator>
    <rdf:Description rdf:about="http://viaf.org/viaf/sourceID/JPG%7C500018431#skos:Concept">
        <rdf:type rdf:resource="http://xmlns.org/foaf/0.1/Agent" />
        <rdf:type rdf:resource="http://schema.org/Person" />
        <schema:name>Ligorio, Pirro</schema:name>
        <schema:birthDate>1500</schema:birthDate>
        <schema:deathDate>1583</schema:deathDate>
        <vra-p:hasCulture>
            <rdf:Description rdf:about="http://purl.org/getty/vocab/ulan/Italian">
                <rdf:type rdf:resource="http://purl.org/vra/Culture" />

                <schema:name>Italian</schema:name>
            </rdf:Description>
        </vra-p:hasCulture>
    </rdf:Description>
</schema:creator>
<schema:creator>
    <rdf:Description rdf:about="http://viaf.org/viaf/sourceID/JPG%7C500010654#skos:Concept">
        <rdf:type rdf:resource="http://xmlns.org/foaf/0.1/Agent" />
        <rdf:type rdf:resource="http://schema.org/Person" />
        <schema:name>Buonarroti, Michelangelo</schema:name>
        <schema:birthDate>1475</schema:birthDate>
        <schema:deathDate>1564</schema:deathDate>
        <vra-p:hasCulture>
            <rdf:Description rdf:about="http://purl.org/getty/vocab/ulan/Italian">
                <rdf:type rdf:resource="http://purl.org/vra/Culture" />
                <schema:name>Italian</schema:name>
            </rdf:Description>
        </vra-p:hasCulture>
        </rdf:Description>
    </schema:creator>
```

## F. RDF Sample Output in Turtle

```
@prefix s: <http://schema.org/> .

@prefix vra-p: <http://purl.org/vra/> .


<#2> a <http://purl.org/getty/vocab/AAT/300007501>,

        <http://purl.org/getty/vocab/AAT/300170443>,

        s:CreativeWork ;

    vra-p:culturalContext <http://purl.org/getty/vocab/ulan/Italian> ;

    vra-p:hasStylePeriod <http://purl.org/getty/vocab/AAT/300021140> ;

    vra-p:material [ a vra-p:Material ;

            s:name "marble" ] ;

    vra-p:placeOfSite [ a s:Place ;

            s:containedIn <http://purl.org/getty/vocab/TGN/1000003>,

                <http://purl.org/getty/vocab/TGN/1000080>,

                <http://purl.org/getty/vocab/TGN/7001168> ;

            s:description "Piazza San Pietro" ;

            s:name """Rome (Vatican City), Santa Sede (Holy See), Italy"""] ;

    vra-p:wasCreated [ a s:Event ;

            s:endDate "1616" ;

            s:name "Creation" ;

            s:startDate "1506" ] ;

    s:about <http://id.loc.gov/authorities/names/n79022118>,

        <http://id.loc.gov/authorities/names/n80030746> ;

    s:creator <http://viaf.org/viaf/sourceID/JPG%7C500007668#skos:Concept>,

        <http://viaf.org/viaf/sourceID/JPG%7C500010654#skos:Concept>,

        <http://viaf.org/viaf/sourceID/JPG%7C500018431#skos:Concept>,

        <http://viaf.org/viaf/sourceID/JPG%7C500019098#skos:Concept> ;

    s:dateCreated "ca. 1506-1616 (creation)" ;

    s:description """Bramante was hired by Julius II to begin a radical

                reconstruction, to replace Old Saint Peter's, and intensive planning
```

```
              by Bramante, Giuliano da Sangallo and Fra Giovanni Giocondo preceded
              the laying of the foundation stone on 18 April 1506. Since Bramante's
              first designs of 1506, half a dozen architects had worked under five
              successive popes, all bringing their own revisions with them. In the
              18 remaining years of his life Michelangelo succeeded in 'uniting
              into a whole the great body of that machine' (Vasari), ensuring that
              the crossing and dome would follow his overall design, even though
              the façade and nave remained unresolved. His new design, as presented
              in the engravings (1569) by Etienne Dupérac, took the form of a
              centralized building over a Greek cross. The façade was built between
              1607 and 1625, and the nave was finally consecrated in 1626; the
              confessio was built to Maderno's design in 1615-1616.""";
    s:hasTechnique <http://purl.org/getty/vocab/AAT/300054608> ;
    s:name "Saint Peter's Basilica"@en,
        "St. Peter's Basilica"@en,
        "San Pietro in Vaticano"@it .

<http://id.loc.gov/authorities/names/n79022118> a s:Person ;
    s:name "Peter, the Apostle, Saint" .

<http://id.loc.gov/authorities/names/n80030746> a s:Person ;
    s:name "Julius II, Pope, 1443-1513" .

<http://purl.org/getty/vocab/AAT/300021140> a vra-p:StylePeriod,
        s:Intangible ;
    s:name "Renaissance" .

<http://purl.org/getty/vocab/AAT/300054608> a vra-p:Technique,
        s:Intangible ;
    s:name "construction (assembling)" .
```

```
<http://purl.org/getty/vocab/TGN/1000003> a s:Continent,
        s:Place ;
    s:name "Europe" ;
    s:url
"""http://www.getty.edu/vow/TGNFullDisplay?find=1000003&place=&nation=&prev_page=1&english=N&subjectid=1000003
""".

<http://purl.org/getty/vocab/TGN/1000080> a s:Country,
        s:Place ;
    s:name "Italy" ;
    s:url
"""http://www.getty.edu/vow/TGNFullDisplay?find=1000080&place=&nation=&prev_page=1&english=N&subjectid=1000080
""".

<http://purl.org/getty/vocab/TGN/7001168> a s:Place ;
    s:name "Rome (Vatican City)",
        "Santa Sede (Holy See)" ;
    s:url
"""http://www.getty.edu/vow/TGNFullDisplay?find=7001168&place=&nation=&prev_page=1&english=N&subjectid=7001168
""".

<http://viaf.org/viaf/sourceID/JPG%7C500007668#skos:Concept> a s:Person,
        <http://xmlns.org/foaf/0.1/Agent> ;
    vra-p:hasCulture <http://purl.org/getty/vocab/ulan/Italian> ;
    s:birthDate "1556" ;
    s:deathDate "1629" ;
    s:name "Maderno, Carlo" .

<http://viaf.org/viaf/sourceID/JPG%7C500010654#skos:Concept> a s:Person,
        <http://xmlns.org/foaf/0.1/Agent> ;
    vra-p:hasCulture <http://purl.org/getty/vocab/ulan/Italian> ;
    s:birthDate "1475" ;
    s:deathDate "1564" ;
    s:name "Buonarroti, Michelangelo" .

<http://viaf.org/viaf/sourceID/JPG%7C500018431#skos:Concept> a s:Person,
        <http://xmlns.org/foaf/0.1/Agent> ;
    vra-p:hasCulture <http://purl.org/getty/vocab/ulan/Italian> ;
    s:birthDate "1500" ;
    s:deathDate "1583" ;
    s:name "Ligorio, Pirro" .

<http://viaf.org/viaf/sourceID/JPG%7C500019098#skos:Concept> a s:Person,
        <http://xmlns.org/foaf/0.1/Agent> ;
    vra-p:hasCulture <http://purl.org/getty/vocab/ulan/Italian> ;
    s:birthDate "1444" ;
    s:deathDate "1514" ;
    s:name "Bramante, Donato" .

<http://purl.org/getty/vocab/ulan/Italian> a vra-p:Culture ;
    s:name "Italian" .
```

## References

Burners-Lee, T. (2009, July 18). Linked Data. Retrieved from http://www.w3.org/DesignIssues/LinkedData.html

Carter R., Double J., Rose-Sander, T., & Webster, M. (2011). The VRA Core survey analysis. Retrieved from http://www.vraweb.org/projects/vracore4/pdfs/VRACoreSurveyAnalysis.pdf

CIDOC. (2009, February 23). The CIDOC conceptual reference model. Retrieved from http://www.cidoc-crm.org/

Coyle, K. (2012). Linked Data tools: Connecting on the web. ALA Editions.

Getty. (2009, June 9). Metadata Standards Crosswalk. Retrieved from http://www.getty.edu/research/conducting_research/standards/intrometadata/crosswalks.html

Glaser, H. H., & Halpin, H. H. (2012). The Linked Data Strategy for Global Identity. *IEEE Internet Computing*, *16*(2), 68-71. doi:10.1109/MIC.2012.39

Good, J. (2011, September 15). How many photos have ever been takem?. Retrieved from http://blog.1000memories.com/94-number-of-photos-ever-taken-digital-and-analog-in-shoebox

Isaac, A., Waites, W., Young, J., & Zeng, M. (2011, November 23). Vocabulary and dataset. Retrieved from http://www.w3.org/2005/Incubator/lld/wiki/Vocabulary_and_Dataset#Art_and_Architecture_Thesaurus_.28AAT.29

LOC. (2007, April 5). VRA Core 4.0 – Element description. Retrieved from http://www.loc.gov/standards/vracore/VRA_Core4_Element_Description.pdf

Miller, E., & Westfall, M. (2011). Linked Data and libraries. *Serials Librarian*, *60*(1-4), 17-22. doi:10.1080/0361526X.2011.556427

OCLC. (2010). How Americans use online sources and libraries. Retrieved from http://www.oclc.org/reports/2010perceptions/howamericansuse.pdf

OCLC. (2011). OCLC releases FAST (Faceted Application of Subject Terminology) as Linked Data. Retrieved from http://www.oclc.org/news/releases/2011/201171.htm

OCLC. (2012a, June 20). OCLC adds Linked Data to WorldCat.org. Retrieved from http://www.oclc.org/news/releases/2012/201238.htm

Patel, M., White, M., Mourkoussis, N., Walczak, K., Wojciechowski, R., & Chmielewski, J. (2005). Metadata requirements for digital museum environments. *International Journal On Digital Libraries*,*5*(3), 179-192. doi:10.1007/s00799-004-0104-x

Schema.org. (2012). Organization of schemas. Retrieved from http://schema.org/docs/schemas.html

Singhal, A. (2012, May 16). Introducing the knowledge graph: things, not strings. Retrieved from http://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html

VRA. (2011). Frequently Asked Questions. Retrieved from http://www.vraweb.org/projects/vracore4/vracore_faq.html

VRA. (2007). VRA Core 4.0 Introduction. Retrieved from http://www.loc.gov/standards/vracore/VRA_Core4_Intro.pdf

W3C. (2012a, May 16). Interview: BBC on publishing and Linked Data. Retrieved from http://www.w3.org/QA/2012/05/interview_bbc_on_publishing_an.html

W3C. (2012b). SXLT tutorial. Retrieved from http://www.w3schools.com/xsl/