TALLINNA ÜLIKOOL

OSLO AND AKERSHUS
UNIVERSITY COLLEGE
OF APPLIED SCIENCES

UNIVERSITÀ DEGLI STUDI DI PARMA

Education and Culture DG
ERASMUS MUNDUS

Jennifer Lea Whisler

Analysis of the Usability of a Preliminary Design for an EDM Metadata Mapping Tool

Master thesis
International Master in Digital Library Learning
2012

ABSTRACT

Europeana aims to bring together metadata for cultural heritage objects from institutions throughout Europe, to increase their visibility and accessibility. This project requires that metadata created and stored in a multitude of different formats and variations to be normalized into a single, standard format, which is soon to be EDM. This constitutes a massive effort on the part of institutions and aggregators, and any system that can ease the process of converting millions of metadata records could be very beneficial to these cultural institutions. This research explores the usability of a potential design for a metadata mapping tool intended to assist in the creation of a mapping specification from a local schema to Europeana's EDM format. The design incorporates five components necessary to consider when creating a quality mapping of elements. In order to ascertain usability, a prototype system was created, and a cognitive walkthrough was conducted to identify usability issues. While the design could be a viable option, usability issues must first be addressed, including how metadata information is presented and how the tool handles complex mapping situations.

Master thesis
International Master in Digital Library Learning
2012

Acknowledgements

First, I would like to thank the European Commission for the funding of my study through the Erasmus Mundus Program, and my supervisor, Nils Pharo, for your support and guidance.

Thank you to all of our teachers and guest lecturers; to Kersti Ahrèn Heløe and to Frederico Monaco, for kindly and diligently ironing out all of the logistics of moving 17 students (from 16 countries) around Europe.

Thank you to my family back home, for always believing in me and my dreams.

Deepest thanks to my DILL family; it has been a wonderful two years learning with and from you. You have been a pillar of support and joy during this time so far from home.

And most especially thanks to Rasmus Thøgersen, for being the most unexpected and pleasant surprise I encountered; for your unwavering confidence in my abilities – even when I was in doubt of them; and for all the other reasons you already know. *Tak skal du have!*

Table of Contents

List of Figures

List of Tables

List of Abbreviations

DC – Dublin Core

ESE – Europeana Semantic Elements

EDM – Europeana Data Model

MODS – Metadata Object Description Schema

METS – Metadata Encoding and Transmission Standard

MARC – MAchine-Readable Cataloging

MINT – Metadata INTeroperability Services

HCI – Human-Computer Interaction

OAI – Open Archives Initiative

OAI-PMH – Open Archives Initiative – Protocol for Metadata Harvesting

ORE – Object Resource and Exchange

PHP – PHP: Hypertext Preprocessor

SKOS – Simple Knowledge Organization System

XML – Extensible Markup Language

XSLT – Extensible Stylesheet Language Transformations

Chapter 1: Introduction

1.1 Background

With the significance of data exchange, integration, and reuse, the interoperability of metadata is of upmost importance, especially in the information community (Chan & Zeng, 2006). More and more, institutions are investigating ways to allow the repurposing of previously created metadata in new and diverse contexts. One method of reuse is the accumulation of multiple metadata collections from various sources into a single repository; however, all heterogeneities – both structural and semantic – between the multiple sources of metadata must be addressed to achieve maximum interoperability within such a diverse system. A repository must either implement a highly complex cross-system search or require all metadata be transformed into a single, standard format.

The latter is the approach selected by Europeana, a web portal meant to facilitate the discovery of cultural heritage objects from institutions throughout Europe. Currently, all participating institutions must provide their metadata in Europeana's metadata format, Europeana Semantic Elements (ESE), which is based in Dublin Core, and should sometime in the near future migrate to the Europeana Data Model (EDM), which is a "theoretical data model that allows data to be presented in different ways according to the practices of the various domains who contribute data to Europeana" (Europeana, 2011d, p. 3). While these standards are built upon other well-known schemas (Dublin Core, SKOS, ORE), they are still new and unique, which necessitates the mapping of all metadata provided by institutions to the EDM format.

Millions of metadata records are being contributed to Europeana from cultural institutions throughout Europe, resulting in a huge work effort to map collections of metadata records in local schema into the standard schema for including cultural heritage objects within Europeana. In order to reduce the time and effort required of institutions to perform these mappings, there are several projects underway to create tools to ease and aid in the mapping process. While many tools were created and are in use for the mapping of metadata to the original Europeana schema, ESE, mapping tools for EDM have yet to be fully developed.

1.2 Introduction to Europeana

Europeana is a project of collaboration between cultural heritage institutions throughout Europe. Museums, galleries, archives, and libraries have come together to increase the visibility and accessibility of digital collections of cultural heritage objects. Europeana harvests metadata in their own standardized format for a unified display in their own web portal, as well as for use in other applications. A more complete profile of Europeana is presented in Chapter 3, along with descriptions of the current and forthcoming Europeana standardized formats, ESE and EDM, respectively.

1.3 Introduction to Metadata Mapping Creation

In order to convert institutional records from the format – or metadata schema – in which the metadata is currently stored into Europeana's standard format – be it ESE or EDM – a metadata mapping must be created. The creation of a mapping between two schemas consists of creating sometimes complex connections between the elements of each schema, while taking into account the semantic definitions, structures, constraints and syntax of all the elements (NISO, 2004). This process can be time consuming and complicated, especially if the two schemas are greatly disparate in domain and/or structure. The definition, characteristics, and complications of metadata mapping will be further explored in Chapter 2.

1.4 Statement of the Problem

Europeana has expressed lofty goals as to the number of cultural heritage objects intended to be represented within its portal over the next few years – over 30 million objects by 2015 – with an especial focus on institutions within those countries which are currently underrepresented (Europeana, 2011d). This will require aligning and normalizing the metadata of a multitude of European cultural institution collections to Europeana's soon to be standard format, EDM, in addition to considerations which must be made for the 20 million object records which are already contained in Europeana in the ESE format. In addition, while common metadata schema may eventually have optimal mappings to EDM, the disparities between unique implementations of metadata schemas necessitate individually considered mappings. In order to accomplish this, a vast amount of time and effort must be dedicated to the creation of mappings. Some tools to aid in the creation of these mappings have yet to be fully developed for EDM. In addition, quality mappings require the juxtaposed consideration of schema, element, and instantiation information for both source and target metadata schema (Chan & Zeng, 2006; St. Pierre & LaPlant, 1998), and the mapping tools currently in development for Europeana's EDM do not present these in conjunction.

1.5 Aims & Objectives

This thesis will analyze a proposed mapping tool design, which is meant to incorporate the creation of a prototype mapping tool that will facilitate the construction of a mapping specification from local institution metadata standards to EDM. This tool would aid in the creation of mapping specifications, which outline precisely how the element values of one schema – the source schema – will be mapped to the available elements in EDM – the target schema. This specification can then be used to inform the transformation of the metadata to the new format (e.g., through the use of XSLT). The aim of this thesis is to explore the use of a low-barrier, HTML table-based tool to assist with the mapping of cultural heritage metadata into the future Europeana metadata format, EDM. Objectives include:

- Investigate the first implementation of EDM;
- Develop a prototype tool to assist in mapping to EDM, using an HTML table structure that incorporates source and target element information necessary for a quality mapping;
- Examine frequent mapping situations within the context of EDM as the target schema; and
- Analyze the functionality and usability of the mapping tool prototype design.

1.6 Research Questions

What are the functional requirements for creating a browser-based prototype for mapping cultural heritage schema to the Europeana metadata standard, EDM?

To what extent is the prototype able to complete potential element mapping situations from the source metadata formats into the first implementation of the Europeana Data Model (EDM) format?

What are the usability issues associated with this HTML table-based structure for a mapping tool to create specifications from local cultural heritage collections into the first specification of EDM?

1.7 Justification

Since the launch of the Europeana pilot project, over 20 million records for cultural heritage objects from more than 1500 museums, libraries, galleries, and archives in 32 European countries have been made accessible through its portal, europeana.eu (Europeana, 2011e). As Europeana employs its own unique metadata schema based largely on Dublin Core, all of these records were required to be transformed from their original format into the created schema, ESE. This translates into an enormous number of man hours spent creating mapping specifications and corresponding transformations. A low-barrier tool that can be used by institutions to better understand their collections and more easily create mapping specifications to the new format EDM could potentially provide a reduction in these man hours and perhaps provide mapping specifications more suited for individual collections.

1.8 Research Design

This research is qualitative in nature. The framework for the research has its basis in human-computer interaction (HCI) and software engineering. In order to answer the research question, a prototype of the proposed mapping tool design was created, following the Process of Prototype Development, frequently used in software engineering (Sommerville, 2011). This plan splits the prototyping process into four steps:

1. Establish Prototype Objectives
2. Define Prototype Functionality
3. Develop Prototype
4. Evaluate Prototype

The objectives of the prototype center on the analysis of the usability of the proposed design, as does the prototype functionality. As such, in order to complete the prototype evaluation, a method and criteria for evaluating usability had to be chosen and specified. A version of the cognitive walkthrough – a usability analysis method used in the HCI domain – was deemed the most appropriate method for evaluating how the prototype performed in common metadata mapping situations that might be encountered by users of a mapping tool.

The cognitive walkthrough was especially suited for this usability evaluation, as it focuses on specific, fundamental tasks the user would likely attempt to perform when interacting with a system. The creators of the cognitive walkthrough (Wharton, Rieman, Lewis, and Polson) modified their method twice (Mahatody, Sagar, & Kolski, 2010). The third version is meant to be “flexible enough to fit into given software development process [and] the method identifies problems with a design early in the process and, by describing the reasons for those problems, it suggests design changes early on” (Wharton, et al., 1994, p. 139, quoted in Mahatody, et al., 2010, p. 776). In this version, meant to be an improvement and simplification of the previous versions, there is a preparation phase and an evaluation phase. In the preparation phase, the user and system are defined, and representative tasks are chosen to be evaluated. Then, the tasks are broken down into the individual steps required for task completion. In the evaluation phase, the evaluator critically analyzes the system’s usability through providing answers and justifications for each of the following questions for every action step:

- Will the user try to achieve the right effect?
- Will the user notice that the correct action is available?
- Will the user associate the correct action with the effect they are trying to achieve?
- If the correct action is performed, will the user see that progress is being made toward solution of their task?

(Wharton, Rieman, Lewis, & Polson, 1993, p. 9)

For the purposes of this thesis, slight modifications were made to the method of the preparation phase. First, as this research is meant to examine the usability of the design in mapping situations, these situations were considered first, and representative tasks were created to reflect these situations. Second, each mapping situation was considered in the context of EDM as the target schema in the mappings. The evaluation phase remained unchanged.

1.9 Limitations

Due to time and resource constraints, there have been certain limitations to this research. For instance, the prototype itself is not a fully functioning system. The focus of the prototype development was placed on the mapping tool display so as to test its usability.

EDM began as a theoretical data model, and is now, through what will be the first implementation of the standard, being translated into practical use. This implementation is limited itself, and the prototype is further simplified by only including the elements contained in the first implementation of the three core classes alone, and does not include the contextual classes[1], which could add further complications to usability, as well.

While in practice, cognitive walkthroughs are often undertaken by the developers of the system itself, researchers generally use one or more evaluators separated from the project to avoid potential bias within the evaluation. However, cognitive walkthroughs can be extremely time-consuming, depending on the number of tasks undertaken (in addition to the time required to familiarize the evaluator(s) with the cognitive walkthrough process). Therefore, as the intended number of tasks to be analyzed was so high and there was some precedence in the literature for authors/developers to perform their own cognitive walkthrough, the developer as evaluator was deemed appropriate (Mahatody, Sagar, & Kolski, 2010; Pinelle & Gutwin, 2002). While subjectivity can never be completely avoided, the researcher attempted to be as objective as possible. While only the analysis of the cognitive walkthrough is included in Chapter 6, to present transparency in the research, the entirety of the evaluation phase of the cognitive walkthrough analysis – including all tasks, action steps, question answers and justifications – is contained within Appendix F.

1.10 Outline

This thesis is split into seven chapters:

- Chapter 1 provided an introduction to the research, including the problem, the research questions, the method of research, and limitations of the research.
- Chapter 2 affords a review of relevant literature, providing the framework and context in which the research is contained. This includes an overview of topics including metadata, metadata schema, and the multiple methods of interoperability with a focus on metadata mapping and mapping tools.
- In Chapter 3, a profile of Europeana is presented, along with descriptions of the Europeana metadata formats, ESE and EDM.
- Chapter 4 provides a detailed explanation of the research methodology and methods. Both the process of prototype development and the cognitive walkthrough usability evaluation method are explained in depth, including method justification and explication of any divergence in normal method procedure.

[1] The core and contextual classes of EDM will be explained in Chapter 3.

- Chapter 5, an extension of the previous chapter, puts the prototype process into practice. The functionality and development of the prototype are described.
- Chapter 6 presents the analysis of the cognitive walkthrough process.
- The final chapter includes the conclusions from the analysis in the context of the research questions, a discussion of the findings, and thoughts on potential future research.

1.11 Chapter Summary

The Europeana project, which unites records of digitized cultural heritage objects from institutions throughout Europe, requires millions of metadata records to be mapped to Europeana's metadata format for normalized search and display. This conversion involves a vast effort on the part of the institutions and/or aggregators, which could be partially alleviated through the use of mapping tools. While there are some tools available for mapping to the current standard, ESE, tools for mapping to the upcoming standard, EDM, are still in the development phase. These tools also fail to juxtapose all source and target element information indicated by St. Pierre & LaPlant (1998) and Chan & Zeng (2006) as essential to the mapping process of real world data. Therefore, this research investigates the usability of a possible design that would bring this information together in a tool to aid in mapping metadata collections to EDM. In order to answer the research questions, a prototype of the proposed design was created, and a cognitive walkthrough was conducted to analyze the functionality and usability of the design.

Chapter 2: Literature Review

2.1 Introduction

Europeana, the standards Europeana maintains, and the processes necessary to converge millions of records from thousands of cultural heritage institutions throughout Europe – including tools that ease these processes – all have at their base the core concepts of metadata – the way these records are expressed; schemas – how this metadata is organized; and the many and varied forms of achieving interoperability – how disparate metadata, schemas, and systems are made to interact, combine, and/or share information. The following sections delve deeply into these concepts, exploring definitions and literary discourse concerning their composition and categorization. In addition, various methods of interoperability are presented, with an especial focus on metadata mappings, which are central to the use of mapping tools.

2.2 Metadata

Metadata is often merely explained as 'data about data.' However, this description does little to delve into the depth and breadth of what metadata entails. A more robust definition is that which is provided by NISO (2004, p. 1), and identifies metadata as "structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use, or manage an information resource." As NISO's definition implies, metadata has many different functions, from increasing findability to allowing for organization, interoperability, record uniquification – the ability to uniquely identify an object, both through a distinctive identifier and the sum of the set of metadata – and preservation. NISO also expounds upon common typologies of metadata, adjoining the types of rights management and preservation metadata to the archetypal set of descriptive, structural, and administrative metadata. Table 2.1 includes examples of each metadata type.

Metadata Type	Example
Descriptive	title, author, publication year
Structural	page order
Administrative	creation date
Rights Management	copyright information
Preservation	precise format information

Table 2.1: Metadata Types and Examples

Descriptive metadata describes the resource, providing data like title, author, and publication year. (Zeng & Qin, Metadata, 2008, pp. 7-9; NISO, 2004, pp. 1-2) Structural metadata denotes how portions of the resource fit together, as in the page order of a scanned book. Administrative metadata details backend and technical information about the resource record, like its creation and modification dates. Rights management metadata explains the rights for the resource, e.g., copyright information; and finally, preservation metadata includes, among other metadata, information about the environment required for use, e.g., hardware, software, etc. (Caplan, 2006, p. 13)

This specification of types is one way of delineating metadata. Another approach is to demarcate metadata based on a technological spectrum. Haslhofer & Klas (2010) outline such a categorization, by classifying four sets of metadata in information systems, from low-level to high-level. At the lowest tier, is the *physical* metadata of the hardware – the actual bytes at the core of computer technology. Next, is the *logical* level, where data in relational databases persists. The *programming* or *representation* level is where this data can be manipulated. It encapsulates coding and mark up languages with strict syntax, which are then defined for human comprehension and use on the final level – the *conceptual* level, which includes metadata schema. These various metadata for a specific resource, gathered together into a set of elements creates a metadata instance (Haslhofer & Klas, 2010, p. 7:7). The grouping of elements is often further elaborated within a specific metadata schema.

2.2.1 Metadata Schema

Haslhofer and Klas (2010) define a metadata schema as "a set of elements with a precise semantic definition, optionally connected by some structure" (pp. 7:7-7:8). Schema can further be broken down into two subdivisions – semantics and content (Chan & Zeng, 2006). The semantics of a schema is the definition of each element, explaining what it is. Content guidelines – often provided along with the elements' semantics – outline acceptable values for the element, including standardized language or formatting rules. Authority files for persons and authors; controlled vocabularies; country codes; and date layout requirements are all instances of content guidelines.

Finally, a schema might also provide a specific modeling language at its inception, that is, the machine-readable language in which the schema must be written (see section 2.3). A schema without this constraint would be considered syntax-independent, and can be encoded in the language chosen by the implementing body (NISO, 2004).

There is a plethora of oft-used schema from a wide range of domains that may be used for resource description. MODS, METS, DC, and TEI are a few schema commonly used in the cultural heritage domain, and a brief introduction to these schemas is provided.

2.2.1.1 MODS

MODS, or the Metadata Object Description Schema, is a very rich metadata schema developed from MARC21 (NISO, 2004, pp. 5-6). Instead of the numeric tags and indicators used in MARC, MODS uses language-based tags and is expressed in eXtensible Markup Language (XML), a common encoding language which will be outlined later in this chapter. Nor is MODS bound to the Anglo-American Cataloguing Rules (AACR2) (Miller, 2011, p. 164). Like MARC, MODS provides the opportunity to create rich descriptions for resources, and its derivation from MARC allows for easy carriage of metadata from MARC21, and can be combined with METS – as well as additional schemas – for the description of complex digital objects.

2.2.1.2 METS

METS, the Metadata Encoding and Transmission Standard, is a container schema, which can include different types of metadata and elements from various other schemas (Zeng & Chan, 2006). Realized by the Digital Library Federation, METS, which is also an XML-based schema, is "a method for expressing and packaging together descriptive, administrative, and structural metadata for objects within a digital library…[and] provides a document format for encoding the metadata necessary for management of digital library objects within a repository and for exchange between repositories" (NISO, 2004, pp. 4-5). METS has seven container sections:

- METS Header,
- File Section,
- Structural Map,
- Structural Links,
- Behavior,
- Descriptive Metadata, and
- Administrative Metadata.

The schema is detailed in regards to these sections except for the latter two – descriptive and administrative metadata – which can be supplemented with the use of other schemas, including MODS and simple Dublin Core.

2.2.1.3 Dublin Core

Dublin Core (DC) in its initial formulation was created to provide creators of web resources with a straightforward and concise metadata standard for resource description (NISO, 2004, p. 3). The notion was that professional cataloguers would not be able to provide metadata for the ever-increasing amount of web content, and could not expect content creators to describe their resources in the exceedingly complex standards of AACR2 and MARC. Yet, quality metadata was still required for these resources. Thus, in 1995, the beginnings of the Dublin Core Metadata Initiative (DCMI) were formed by OCLC, which resulted in a "core" metadata set of 15 elements. To increase simplicity, all elements are repeatable and none are required. These elements were later extended with the inclusion of refinements – qualifiers for the original elements to allow for greater specificity (Miller, 2011, p. 49). For example, the original element "Date" could now be qualified with "Created", "Valid", "Available", "Issued", or "Modified" – all of which are allowable value choices for the unqualified element. The intention was to provide the option for deeper semantic meaning, while still adhering to the original DC set. These extensions became known as qualified DC, while the original set is called simple – or unqualified – Dublin Core. However, both qualified and unqualified DC are far less complex than MARC and MODS. In contrast with MODS and METS, no modeling language was specified for use with DC.

2.2.1.4 TEI

The Text Encoding Initiative Guidelines for Electronic Text Encoding and Interchange, known as TEI format, is maintained by the international TEI Consortium (TEI Consortium, 2012). Expressed in XML, TEI is "a markup language for representing the structural, renditional, and conceptual features of texts" (TEI Consortium, 2012). Focused on use in the humanities, TEI is largely used to markup digitized print materials, and therefore provides the capacity to include metadata about both the original resource and the electronic representation (NISO, 2004, p. 4). Since the full TEI specification is extremely complex, TEI Lite – a simplified version – was created.

As is apparent from these examples, metadata schemas are extremely diverse. They are created for various communities, purposes, and domains and at significantly different levels of complexity. An additional consideration for metadata schema is the syntax in which it is presented. Most of the above schemas are tied to and must be expressed using the extensible markup language (XML), and even those that do not specify a modeling language, can and have been expressed in XML. Section 2.3 introduces modeling languages in general, and XML specifically.

2.3 Modeling Languages

A metadata schema itself denotes only the set of allowable elements and their semantic meanings. For a computer to interact with a metadata schema, it must be encoded in a way the computer can understand. This is achieved using a modeling language, which provides a manifestation of the schema (Haslhofer & Klas, 2010). A modeling language has machine-readable syntax, which "refers to the structure of a program and the rules about that structure" (Downey, Elkner, & Meyers, 2008, p. 4). As such, it provides structure to the presentation of the schema, and can be manipulated by computer systems and applications. A very popular modeling language for metadata schema is XML, which is utilized by Europeana. A short introduction to XML is provided below.

2.3.1 Extensible Markup Language (XML)

The markup language XML has become a pivotal unit in data exchange and reuse in the digital age (NISO, 2004, p. 3). Like HTML, XML was derived from the Standard Generalized Markup Language (SGML) (Harold & Means, 2004, pp. 8-10). Unlike HTML, which is simply an application of SGML, XML is essentially a simplified version of SGML, which in its original form was exceedingly complex. Endorsed by the Word Wide Web Consortium (W3C), XML "defines a generic syntax used to mark up data with simple, human-readable tags" (Harold & Means, 2004, p. 3). It is considered a metamarkup language because the tags, or elements, used to markup a document are not predefined – instead, creators of XML documents can define their own elements, as well as elements' attributes. Though no set vocabulary is stipulated, XML specifies strict syntax to ensure documents can be processed. For instance, there are rules on how to indicate a tag and where tags may be placed, and XML requires a single "root" element that contains all others elements. A document which begins with the XML declaration statement and follows all syntax rules is said to be well-formed. A very basic XML document might look like this:

```
<?xml version="1.0">
<note>
      <to>Bea Potts</to>
      <from>Nicholas Tuck</from>
      <heading>Reminder</heading>
      <body>Don't forget me this weekend!</body>
</note>
```

XML is very versatile, and can be used as a foundation format for structuring data that can then be manipulated in many ways, as it is "an incredibly simple, well-documented, straightforward data format" (Harold & Means, 2004, p. 6).

While an XML document need only abide by syntax rules to be well-formed, institutions and communities often provide further constraints as to what elements may be used and in what way. This can be done using Document Type Definitions (DTD) or XML schemas. A DTD can be included within an XML document, or be a separate document that is referenced after the XML declaration. The DTD identifies allowable elements; these elements' sub-elements; how many times the element may appear; allowable attributes; and possible content of the elements. A DTD might look like this:

```
<!ELEMENT NEWSPAPER (ARTICLE+)>
<!ELEMENT ARTICLE (HEADLINE,BYLINE,LEAD,BODY,NOTES)>
<!ELEMENT HEADLINE (#PCDATA)>
<!ELEMENT BYLINE (#PCDATA)>
<!ELEMENT LEAD (#PCDATA)>
<!ELEMENT BODY (#PCDATA)>
<!ELEMENT NOTES (#PCDATA)>

<!ATTLIST ARTICLE AUTHOR CDATA #REQUIRED>
<!ATTLIST ARTICLE EDITOR CDATA #IMPLIED>
<!ATTLIST ARTICLE DATE CDATA #IMPLIED>
<!ATTLIST ARTICLE EDITION CDATA #IMPLIED>
```

Another option for specifying the construct of XML documents is the use of an XML schema. While the general usage of XML schemas is similar to that of DTD's, schemas are themselves well-formed XML documents. An example XML schema is below.

```
<?xml version="1.0">
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="order">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="person" type="xs:string"/>
      <xs:element name="customer">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="name" type="xs:string"/>
            <xs:element name="address" type="xs:string"/>
            <xs:element name="city" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="item" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="title" type="xs:string"/>
            <xs:element name="quantity" type="xs:positiveInteger"/>
            <xs:element name="price" type="xs:decimal"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="id" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>
</xs:schema>
```

A well-formed XML document that also follows all the rules set forth in its referenced DTD or XML schema is said to be valid. This allows for an XML document both to be internally consistent, and to meet criteria for use in specific contexts. Another powerful feature of XML is the ability to create and call upon namespaces. A namespace is “a way to tie a specific use of a metadata element to the scheme where the intended definition is to be found” (NISO, 2004, p. 16). Use of namespaces disambiguates between elements and attributes that may have the same name, providing a specific context for the elements’ use (Harold & Means, 2004, p. 60). In addition, namespaces can group together all elements and attributes of a particular application. The previously mentioned metadata schemas, when expressed in XML, each have unique Uniform Resource Identifiers (URI) which are utilized as namespaces to identify XML elements as members of their schema’s element set. The following is an example employing the Dublin Core namespace:

```
<?xml version="1.0">
<record xmlns:dc="http://purl.org/dc/">
      <dc:title>The Illiad</dc:title>
      <dc:creator>Homer</dc:creator>
      <dc:description>From bn.com: "The epic song of
Ilion (an old name for Troy), The Iliad recreates a few
dramatic weeks near the end of the fabled Trojan War,
ending with the funeral of Hector, defender of the doomed
city."<dc:description>
</record>
```

The consistency of syntax and ease of use of XML is very beneficial in the sharing of metadata. Protocols like OAI-PMH build upon this base by providing a method for organizations to uniformly expose XML metadata.

2.4 Protocols

According to Gerard & Singh (2010), "protocols are a way to standardize communication patterns so agents can be used in many different multiagent interactions" (p. 1). Protocols provide an exact framework through which systems communicate and work together (Zimmermann, 1980). This allows for communication between systems within a vast network. For instance, Hypertext Transfer Protocol (HTTP), Transmission Control Protocol (TCP), and Internet Protocol (IP) are standard protocols used for the internet. The Open Systems Interconnection (OSI) model distinguishes between seven layers of protocols; protocols in the lower layers – starting with the physical layer – are established and built upon to the higher layers – ending with the application layer (Zimmermann, 1980). OAI-PMH, which is discussed below, is the application layer protocol for harvesting metadata used by Europeana.

2.4.1 OAI-PMH

Originally concerned with the swift and easy sharing of e-print publications from various sources, the Open Archives Initiative (OAI) began with the desire to find "alternatives to the traditional scholarly publishing paradigm" (Lagoze & Van de Sompel, 2001, p. 54). The initial focus was on interoperability between the various sources of e-prints metadata in order to enable sharing and potentially unite this information with greater ease. The Initiative decided upon metadata harvesting as the preferred method for interoperability, wherein providers would expose their metadata in an open and uniform manner for others to access and reuse, or harvest. The intention was to keep the method low-barrier, i.e. ensure that implementation and use of the interoperability standard would be as simple as possible to increase the likelihood of widespread use. As the OAI plans for metadata harvesting progressed, communities outside of the e-print world became interested in implementing such a harvesting protocol, and the trajectory of the standards shifted toward a more inclusive, less specified approach. As such, the technical framework for the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) evolved, was tested, and was eventually implemented. The framework delineates two types of participants: data providers and service providers. As the names suggest, the former expose their data and the latter harvest this data – both using the specified protocol.

The protocol distinguishes between three different entities in relation to the provided metadata – resource, item, and record (Lagoze, Van de Sompel, Nelson, & Warner, 2002). A resource is the actual entity that the metadata describes. The resource may be electronic and may be directly linked in the metadata, but this is not required. A resource can also be physical in nature. The metadata representation of this resource in the repository is called an item, and the presentation of this metadata in a specific format is called a record. This separation between item and record is important, as the framework allows for metadata to be presented in multiple formats. Therefore, though each resource has only one corresponding item within the repository, there could potentially be multiple records presenting the same data in different ways. Every item in the repository has a unique OAI identifier, and all records corresponding to this item share this identier. The OAI protocol allows for the use of multple formats for the representation of metadata, as long as the encoding language is XML; however, unqualified Dublin Core is the only required metadata format.

The focus of the technical framework for data providers is the uniformity of how records must be presented upon request. A record returned over OAI-PMH consists of at least two sections, the *header* and the *metadata* sections (Lagoze, et al., 2002). The header hold the unique OAI identifier, the datestamp (which could indicate the date of creation, modification, or deletion), and a setSpec element for each set to which the item belongs, if any. The metadata section indicates the format used for this particular representation of the item and includes all metadata. The final, optional *about* section allows for the inclusion of information concerning the metadata, e.g. the rights statement for use of the metadata. This consistency allows for all metadata accessed via OAI-PMH to be used and potentially manipulated similarly.

For service providers, the framework explicates how data may be accessed via HTTP GET or POST requests (Lagoze, et al., 2002). OAI requests begin with the repositories base URL, determined by the server's internet host, port, and path. This base URL is then followed by one or more keyword arguments, the first of which is always the OAI verb key, which has six allowable requests (Zeng & Qin, 2008, p. 228). Additional keyword arguments may – and in certain cases, must – be used to further constrain retrieved results. An example request is shown in Figure 2.1.This request will return a list of the metadata records in the 'billeder' collection, and the records will be displayed in the MODS metadata schema.

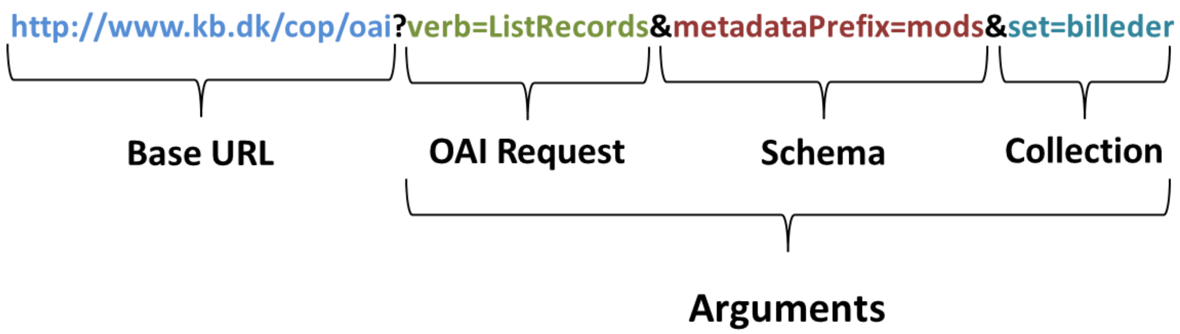

Figure 2.1: Sample OAI-PMH Request

As previously stated, Europeana almost exclusively uses OAI-PMH to harvest metadata from contributing institutions. Sharing and harvesting of metadata via OAI-PMH, is one of many methods to achieve greater interoperability.

2.5 Interoperability

Metadata interoperability is defined as "a qualitative property of metadata information objects that enables systems and applications to work with or use these objects across system boundaries." (Haslhofer & Klas, 2010, p. 7:14) Interoperable metadata should be usable in multiple ways and serve diverse functions. With the ever-growing significance of data exchange, integration, and reuse in the digital age, the interoperability of metadata is of upmost importance, especially in the information community (Chan & Zeng, 2006, p. 3). For instance, among NISO's (2007) list of quality metadata requirements in *A Framework of Guidance for Building Good Digital Collections*, is the stipulation that "Good metadata supports interoperability" (p. 61). Even carefully created metadata, therefore, lacks quality if it cannot be used outside its original context. As Intner, Lazinger, & Weihs (2006) state:

> "Given the choice between a perfect but unique metadata schema utterly lacking in interoperability and a moderately good schema that gets high marks for interoperability, most experts recommend the latter. In part, this is because digital libraries are, by their very nature, better when they are bigger and more diverse, and

> to get as big and diverse as possible, it takes more than one institution, library, or other data generating entity. The likelihood that the originators of a digital library project intend to be the sole contributors to it for all time is minuscule compared with the likelihood they will seek partners. In a collaborative environment, interoperability trumps perfection every time." (p. 189)

When metadata may eventually be used in heterogeneous information systems, interoperability is simply essential, and, in the increasingly digital world, interoperability must be a central criterion for all metadata (Haslhofer & Klas, 2010, p. 7:1; Yuan, Bahrami, Wang, Murray, & Hunt, 2006, p. 1171). Therefore, interoperability must be considered at the outset of a metadata project.

While interoperability can be discussed on any of the metadata levels defined by Haslhofer & Klas[2] (2010, p. 7:9) – physical, logical, programming/representation, and conceptual – this project assumes interoperability at lower levels, including the use of the same modeling language (XML) at the programming/representation level. Therefore, this discussion will focus mainly on the conceptual level of metadata. Interoperability itself can also be discussed at multiple levels, creating an opposing axis to the levels of metadata. The different modeling of these levels is expounded below.

2.5.1 Levels of Interoperability

Interoperability can be viewed as a pyramid of levels, where lower levels of interoperability are necessary, though not alone sufficient, to achieve maximum interoperability between systems. Various models have been presented to explicate these levels.

One example is the Levels of Information Systems Interoperability (LISI), part of the U.S. Department of Defense Architecture Framework, which includes five levels of systems interoperability: Isolated, Connected, Functional, Domain, and Enterprise (U.S. Department of Defense, 1998, pp. 2-5).

With a focus on the data exchange process, Tolk & Muguira (2003) identify the Levels of Conceptual Interoperability Model. The levels include System Specific Data, Documented Data, Aligned Static Data, Aligned Dynamic Data, and Harmonized Data. Figure 2.2 further illustrates the functional interoperability at each level for the two models.

[2] As outlined in Section 2.2

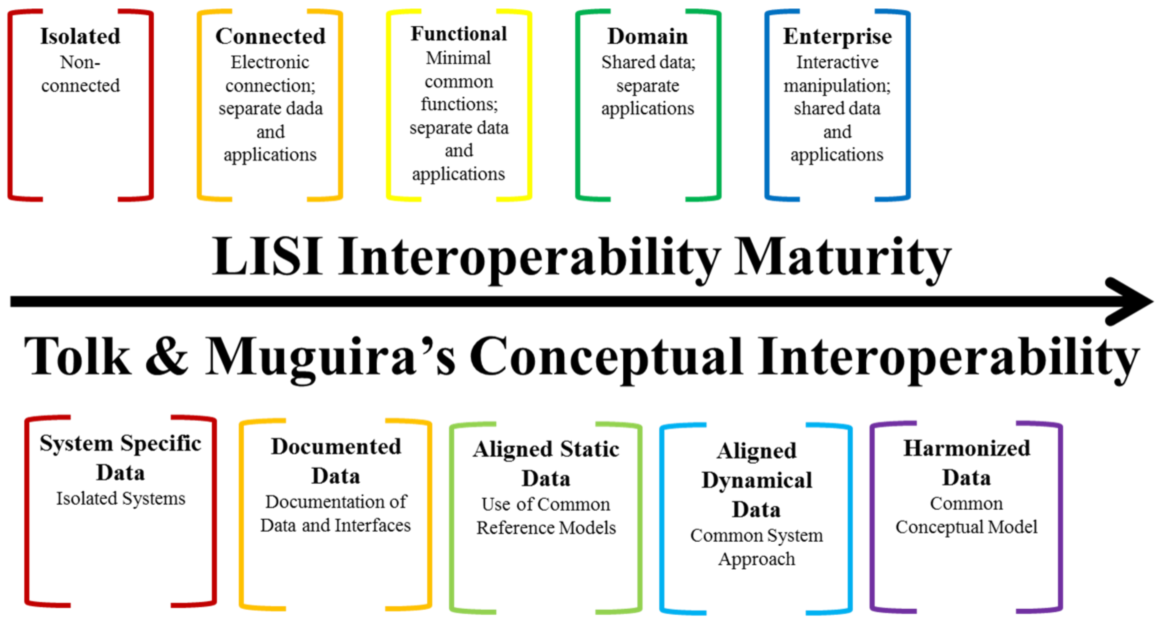

Figure 2.2: Representation of Interoperability Models **(U.S. Department of Defense, 1998; Tolk & Muguira, 2003)**

For both models, the lowest level, Level 0, is indicative of no interoperability, where there is no exchange of data. As the levels increase, functionality increases from data accessibility, into greater and greater degrees of interoperability until the systems are fully interoperable. Tolk & Muguira argue that the final level of their model, the Harmonized Data Level, outlines what cannot be achieved solely through the technical interoperability encompassing models like LISI. Fully semantic interoperability, which they state is required for finding "quasi-optimal solutions to real-world problems," necessitates a conceptual, not technical, view where semantic relationships between elements are examined (pp. 5-6). In the digital library context, these levels could be simplified to the object level, service level, and semantic level (Vullo, Innocenti, & Ross, 2010, p. 39).

Another approach is separating interoperability implementation based on methodological considerations – that is, based on the current complexity of the data in question. Chan & Zeng (2006) identify three methodological levels – schema, record, and repository – in their characterization of interoperability techniques. Schema level interoperability applies prior to the creation of any metadata records; record level when records already exist and must become interoperable with one or more new systems; and the repository levels when metadata will be joined from many heterogeneous systems. Each level presents its own challenges and potential solutions, though Chan & Zeng emphasize these are not mutually exclusive to a specific level. These solutions will be explored in Section 2.5.3 as methods to achieve interoperability.

2.5.2 Challenges to Interoperability

All challenges encountered when attempting to achieve interoperability arise from a single difficulty: variance. Whether at the lowest technical level, or the highest semantic level, any variance can result in an interoperability conflict, which necessitates a solution.

Potential deterents to interoperabiliaty are numerous. Spaccapietra, Parent, & Dupont (1992) classify four types of interoperability conflicts (pp. 89-90). The first is *semantic conflict*, where two models have “overlapping sets” of objects, where either an object in one model is a subset of an object in the second model, or the objects partially intersect. *Descriptive conflicts* arise when different factors are used to describe the same set of objects. *Heterogeneity conflict* occurs when different data models are used. Finally, *structural conflicts* occur when the same object is formatted in different ways. For instance, in XML, the same data may be expressed in multiple ways, using attributes or sub-elements:

```
<title type=“main”>The Lord of the Rings</title>
<title type=“subtitle”>The Two Towers</title>

<title>
<main>The Lord of the Rings<main>
<subtitle>The Two Towers<subtitle>
</title>
```

In the first example, the title becomes two *title* elements. The attribute *type* is utilized to distinguish between the main title and the subtitle. In the second example, the title is split into two unique elements, *main* and *subtitle,* which are both subelements of the element *title*.

It is clear that in order for interoperability to be reached, data must be consistent, therefore discrepancies between data representation must be identified and addressed (Tolk & Muguira, 2003, p. 2). Focusing on these discrepancies between metadata schema on the element and instance level, Haslhofer & Klas (2010) define eleven potential heterogeneity classes:

Structural Heterogeneities – model incompatibilities

- *Naming Conflicts* – different element names for the same object
- *Identification Conflicts* – one schema’s elements can be uniquely identified, while the other’s elements cannot
- *Constraints Conflicts* – schemas define element constraints differently
- *Abstraction-Level Incompatibilities* – elements are at differing levels of granularity

- *Multilateral Correspondences* – an element in one schema corresponds to multiple elements in the other
- *Meta-Level Discrepancy* – an element in one schema lacks an absolute equivalent element in the other, though close equivalencies may still be possible
- *Domain Coverage* – an element in one schema has no equivalent in the other; the real-world entity is modeled by the element is absent from the other model

Semantic Heterogeneities – differences in the domains and element definitions of schemas

- *Domain Conflicts* – the models do not cover precisely the same domain
- *Terminological Mismatches* – synonyms and homonyms
- *Scaling/Unit Conflicts* – use of different units for element values
- *Representation Conflicts* – use of different encoding standard for element values

(Haslhofer & Klas, 2010, pp. 7:14-7:17)

While many in the above list should be solved by utilizing the same schema in a carefully defined domain, even this does not remove all interoperability conflicts. In the creation of a cross-institutional digital repository, for example, Costanza, Knight, & Liu-Spencer (2009) found heterogeneities even among institutions implementing the same schema – Dublin Core – and in very similar domains – a consortium of U.S. liberal arts universities (p. 156). These differences stemmed both from the inconsistent use of specific elements by participating institutions (description, rights management elements) and differing implementation standards for element content values (controlled vocabulary or lack thereof). Paepcke, et al. (1998) maintain that such a collective effort should strive for both independent growth and interoperability (p. 33). Yet, while this may be the "ultimate goal" of such a heterogeneous, multi-institutional project, the complexities of aggregating metadata from separate, autonomous systems clearly deter the creation of a single, harmonious entity.

However, some compromises can be made, and solutions may be simpler if a certain amount of heterogeneity between aspects of systems is allowable (Paepcke, et al., 1998, p. 36). For instance, if the user interfaces of different collections need not be uniform or specific areas allow for greater functionality than others, systems may still be interoperable while not completely consistent. Nevertheless, for data to be used outside of its original context, often conflicts must be resolved. Currently, there is a multitude of methods for achieving metadata interoperability.

2.5.3 Methods to Achieve Interoperability

Paepcke, et al. (1998) list five methods for interoperability: strong standards, families of standards, external mediation, specification-based interaction, and mobile functionality (p. 39). However, as their approach is a broad overview of interoperability in information systems, most of these methods focus on lower, more technical levels of interoperability, as opposed to the conceptual level. The most relevant to metadata schema interoperability is the administration of strong standards.

Perhaps the most obvious approach, and according to Chan & Zeng (2006) the best, would be for all systems to adhere to a strong, uniform standard. As Tolk & Muguira (2003) state, "the definition of a common ontology and introducing standardized shared data elements has been the topic of various contributions to the interoperability discussions" (p. 2). The advantages of this choice are obvious; if all systems follow the same standards in the same way, there is less chance of divergence, i.e. interoperability is inherent between the systems. Standards agreement can be broken down into language agreement, metadata schema agreement, instance-level agreement, or hybrid metadata systems (Haslhofer & Klas, 2010, pp. 7:17-7:21). With a standard language, all resultant metadata can be processed through similar applications. A standard schema allows for all metadata to be described by the same set of elements. Instance-level agreement manifests in controlled vocabularies, authority control, and value encoding schema for content values; and hybrid metadata systems combine several levels of agreement.

Unfortunately, a standards solution is not always possible to achieve. First, different domains and communities have varied metadata needs, so a single standard may not be feasible between them; and attempts to serve a multitude of needs can lead to standards that are either overly complex or semantically weak (Paepcke, et. al, 1998, pp. 39-40; Haslhofer & Klas, 2010, p. 7:21). This also makes standard development a lengthy process. In addition, though providing greater interoeprability potential, stringent standards limit possibilities for "local optimizations" (Paepcke, et. al, 1998, p. 40). Finally, metadata already exist in various schemas and formats, requiring further interoperability methods to adhere to a new standard. However, in certain contexts and domains, uniform standards can be achieved, especially for interoperability at the schema level, prior to the creation of metadata records.

When a uniform standard cannot be achieved, there are multiple other options for interoperability at the schema level, including derivation, application profiles, crosswalking, switching-across, metadata frameworks, and metadata registries (Chan & Zeng, 2006).

Derivation is an option that allows for higher autonomy than uniform standards, while still ensuring a similar structure and basic elements. In derivation, "a new schema is derived from an existing one" (Chan & Zeng, 2006). Though many variants will ensue, the potential for interoperability between the original schema and the new schema are much higher than had the latter been created separately.

At Chan & Zeng's schema level and Haslhofer & Klas's metamodel agreement level, an **application profile** provides a higher level of homogeneity than derivation for a domain community. Similar to a derivation, an application profile extends and/or explicates an existing schema specifically for a certain community or domain, and can include element definition refinement and allowable element values (Chan & Zeng, 2006; Haslhofer & Klas, 2010, p. 7:23). For instance, the Dublin Core Metadata Initiative "encourages the adoption of application profiles (domain-specific rules) for particular domains" (NISO, 2004, p. 3). Application profiles are made for reuse within specific communities (Haslhofer & Klas, 2010, p. 7:23). Chan & Zeng (2006) reiterate the creation of an application profile, which consists first of choosing a main schema and particular elements from other relevant schema that will be utilized (Chan & Zeng, 2006). These can be further extended with newly defined elements which will then be regulated by the creator(s). NISO (2004) presents a slightly differing view, distinguishing between applications profiles and **extensions** of a metadata schema (p. 9). In their interpretation, profiles can only be a subset of a particular schema, through specifying which pieces of the schema will be used and rules for implementation. An extension, which adds elements to a schema, as does the process outlined by Chan & Zeng, is then considered a separate entity.

When dealing with metadata schemas, **crosswalks, or mappings,** are more prevalent than any other interoperability method and can be used at the schema, record, and repository level (Chan & Zeng, 2006). It is also the method used by Europeana contributors to convert local metadata into Europeana's standard schema, and will be addressed in more detail in the following section.

An extension of crosswalks that can be used when multiple schemas must all interact together is **switching-across** (Chan & Zeng, 2006). Instead of creating bilateral crosswalks between each of the involved schemas, which would be extremely labor-intensive, a single schema is chosen as the “switching mechanism.” Therefore, every schema is merely mapped to this central schema, where it can then interact with any of the schemas in the grouping. Any new schemas need only be mapped once, regardless of how many schemas are in the set.

A **metadata framework** creates basic guidelines in which schemas may be situated (Chan & Zeng, 2006). Either developed prior to the schemas it will encapsulate or derived from and formed around existing schemas, “a metadata framework [provides] a suitable environment for the diverse audiences of involved communities” (Chan & Zeng, 2006).

The final interoperability technique of use prior to metadata creation is a **metadata registry** (Chan & Zeng, 2006). A registry collects specifications for multiple schemas into a central location. This makes for simpler identification of existing schemas that might be used fully or as a basis to model a new schema.

At the record level, metadata that has already been created must be dealt with, and potential interoperability methods for multiple databases or systems are essentially limited to some form of conversion or integration of data values (Zeng & Chan, 2006). Conversion transforms metadata from one schema to another, through the use of a crosswalk. However, a major limitation to conversion is the potential loss of data, because crosswalks generally deal only with schema specifications; the distinctive implementations are not addressed; and real data is not always carefully considered.

Data integration is when metadata from various sources is combined for reuse in different contexts (Zeng & Chan, 2006). The source metadata can be both computer-generated and manmade, and merged together to form a single, rich metadata record for a resource. METS, the previously mentioned container schema, provides the potential for achieving this integration.

When accumulating multiple metadata collections from various sources into a single repository, possible approaches include cross-system search, metadata harvesting, and aggregation (NISO, 2004, p. 2; Zeng & Chan, 2006).

Cross-system search involves searching multiple sources through a single search engine, which is problematic if the systems are heterogeneous as “results are rarely presented in a consistent, systematic, or reliable format” (Zeng & Chan, 2006). To overcome this issue, metadata can be harvested, and either kept in the original format or converted to a single format.

For **metadata harvesting**, repositories may use the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH). However harvesting is performed, if metadata is not converted, consistency of cross-collection search will still need to be realized (Zeng & Chan, 2006). However, this is possible, as demonstrated by the Digital Library for Earth System Education (DLESE) Collection System, which allows for unique XML schema to be defined within their general framework. On the other hand, if a standard format for all metadata in the repository is chosen, the process of conversion must be address for each unique source format.

Similar to data integration, but on the repository scale and tangential to metadata harvesting is the concept of **aggregation**. Instead of merely bringing metadata from multiple sources together, repositories can also "add value" through compiling metadata from different sources for individual resources, thereby creating a "more complete profile" (Zeng & Chan, 2006). The newest iteration of European schemas, the Europeana Data Model (EDM), aims to accomplish this, not only through the aggregation of metadata from contributing institutions, but also through the exploitation open data on the web.

Haslhofer & Klas (2010) provide a simplified grouping of interoperability techniques, by categorizing methods into model agreement (the use of the same metadata standards), metamodel agreement (the linkage of an individual standard to a common standard), and model reconciliation (normalizing heterogeneous models as necessary) (pp. 7:17-7:28). Of these, Haslhofer & Klas argue that methods for model reconciliation are the most desirable and practical. While complete model agreement would be preferable, it is rarely feasible on a large scale. Though the most technically complex, reconciling heterogeneity through metadata mapping is currently the best solution for the diverse and open environment found on the web today (Haslhofer & Klas, 2010, p. 7:33).

2.6 Mappings and Crosswalks

Within the scope of metadata interoperability literature, there arises a terminological inexactitude in the definitions of and interaction between crosswalks and mappings. NISO (2004), for example, subsumes metadata crosswalk within mapping in their definition: "a mapping of the elements, semantics, and syntax from one metadata scheme to those of another" (p. 11). However, no precise definition of "mapping" is provided by NISO, therefore the exact nature of the relationship between the terms cannot be determined. Woodley (2008) provides a more explicit association between the terms, describing a mapping as the "intellectual activity of comparing and analyzing two or more metadata schemas" and a crosswalk as "the visual and textual product of the mapping process." This is substantiated by Kolaitis's (2005) definition of "schema mappings" as "specifications that describe the relationships between schemas at a high level [which] are typically given in a logical formalism that captures the interaction between schemas at a logical level without spelling out implementation details relevant to the physical level" (p. 61). As such, mappings could be considered a conceptual model for the practical application of the crosswalk, though Woodley's definition of the latter does not expressly state that crosswalks are inclusive of the technical specifications required to complete the instantial transformation from source schema to target schema.

Haslhofer & Klas (2010) construct a more complex configuration of the terms – mapping is viewed on multiple levels, and the unique iterations of schemas are taken into account. Mapping itself is more robust, not limited merely to the conceptual/schema level, but is, in effect, any effort to bridge the heterogeneities persistent at any metadata level though source and target element matching. They further extrapolate that mappings must be completed for lower levels – for instance, if the schemas are maintained in different encoding languages – prior to attempting to map between higher levels. While adhering to NISO's definition of a crosswalk and denoting a metadata crosswalk to be a "special kind" of a schema mapping, the terms are addressed as near-equivalence in their discussion of model reconciliation in the digital library domain. In addition, their conceptualization also includes the distinct terms metadata mapping and instance transformation (pp. 7:24-7:25). Instance transformation deals with the manipulation of element *values* to achieve interoperability, and, along with schema mapping, makes up metadata mapping: "Given two metadata schemes, both settled in the same domain of discourse and expressed in the same schema definition language, we define metadata mapping as a specification that relates their model elements in a way that their schematic structures and semantic interpretation is respected on the metadata model and on the metadata instance level" (Haslhofer & Klas, 2010, p. 7:25). Thus, mapping contains both metadata mapping and language mapping, the latter of which is a prerequisite for the former should schema encoding languages be different; and metadata mapping is composed of schema mapping (slightly broader than metadata crosswalk) and instance transformation.

While these variations in semantics may be minor, the greatest inconsistency lies in which term garners the greatest focus. Chan & Zeng (2006), for example, emphasize crosswalk as the most widely spread metadata interoperability technique, and, along with NISO, consider crosswalks to be an asset for unified searching of multiple metadata sources; for reuse of metadata in unanticipated contexts; and for enabling heterogeneous metadata to function homogenously (NISO, 2004, p. 11; Chan & Zeng, 2006). Others concentrate largely – or solely – on mapping, with little or no mention of the term "crosswalk." Even in Haslhofer & Klas's "Survey of techniques for achieving metadata interoperability," "crosswalk" is mentioned as little more than an aside (2010).

An important factor in the quality and ease of creation of a crosswalk is the relationship between the schemas being mapped (NISO, 2004, p. 10). The more similar the schemas' structures and the more closely relatable element definitions, the easier the crosswalk creation will be, and the higher the threshold for the potential quality of the mapping. Also, the intended domains of the source and target schemas have a bearing on the resultant mapping, as well. Clearly, if two schemas cover completely different domains, there is little chance to make semantically significant mappings between individual elements.

According to Chan & Zeng (2006) metadata mapping can either be absolute or relative. In an absolute mapping, the source and target elements must match exactly, or very nearly so. A relative mapping does not require precise equivalence between source and target elements, the main focus is to ensure that all elements from the source schema are mapped to at least one element in the target schema. While semantic context will be lost, relative mapping minimizes data loss from conversion.

In order to create quality mappings between schemas – be they relative or absolute – there are certain important factors that must be addressed for all of the elements on both sides of the mapping. In a White Paper[3] developed for NISO, St. Pierre & LaPlant (1998) identify element components necessary in the creation of a metadata crosswalk. For every element, there should be present:

- A semantic definition;
- Occurrence constraints detailing whether the element is required and/or repeatable;

[3] According to the NISO website, "White Papers are often developed as a pre-standardization activity to define and explore some of the questions that come into play before formal standardization work is started. Or a NISO White Paper might identify areas that are opportunities for standards development and suggest possible approaches" (NISO, 2012).

- Structural constraints detailing relations with other elements, such as a sub-element/super-element relationship; and
- Value constraints detailing what type of information the element can contain, and in what form.

These components provide vital information required to ensure the mapping is valid. Semantic definitions allow the mapper to compare source and target element utilization and meanings – identifying how each source element is used and then attempting to match this use to a target element (or elements).

While this can be challenging enough, it is not the end of the essential considerations. Any constraints upon either the source or target elements being mapped will also affect the mapping. Occurrence constraints outline how often a specific element can or must be present in each record. In unidirectional[4] mappings, occurrence constraints are especially important for the target elements, as the creator of the mapping must ensure that all required elements are included in the mapping and no non-repeatable elements have multiple occurrences; otherwise the mapping will be invalid. When the chosen target element is non-repeatable, the repeatability of the corresponding source element must also be examined. Mapping an element that repeats within even one source record to a target element that does not allow repetition would result in an error when metadata is converted. In bidirectional[5] mappings, occurrence constraints must be carefully examined for all source and target elements.

Structural constraints present how an element is connected to other elements within the schema hierarchy. Deep metadata schemas, like MODS, will have many parent-child element relationships. The parent dictates the nature of the child and must be considered in a mapping. For example, in MODS, *name* can be used as a top level element or as a sub-element of *subject*. An author of the resource would be placed in the former, while a person who is a subject of the resource would be in the latter. This is informed by the structural constraints of the element.

Finally, value constraints limit the allowable values of an element. Again, in unidirectional mappings, this is vital for the target elements, especially. If a target element has specific requirements on how its value may be presented (be it formatting specifications or a controlled vocabulary), either the values of the source element(s) mapped to it must meet these criteria or the mapping must specify how these values will be modified to do so.

[4] When one schema – the source schema – must be mapped to a second schema – the target schema.
[5] When two schema are mapped to each other, allowing for metadata to be transformed from one to the other.

Chan & Zeng (2006) indicate another important factor that often is not adequately considered in mapping creation is the actual data that will be converted. They emphasize that “The reality is that crosswalks constructed based on the real data conversion might be very different from those based on metadata specifications…Unfortunately, most crosswalks are focused only on mappings based on metadata specifications, not on real data conversion results” (Chan & Zeng, 2006). They also state that some element mappings require explanations to extrapolate the mapping choice or more exact instructions on how to complete the mapping. For instance, a mapping may combine two source elements into a single target element, and there must be instructions on how the concatenation of the values will be completed. All of these element components are important to consider when creating mappings, as they can create and/or help in the identification of complex mapping situations.

2.6.1 Mapping Situations

Haslhofer & Klas (2010), in a reiteration of Spaccapietra, Parent, & Dupont (1992), identify four potential mapping expressions which delineate all possible relationships between two elements in different schema.

- Elements can be **equivalent**, where they have the same semantic meaning (whether or not the element names are identical). For example, two schema may both have *author* elements, and both are defined similarly.
- One element can be completely **inclusive** of the other. For instance, a subtitle element in MODS would be included within the title element of simple Dublin Core.
- The elements can **overlap**, so that in some cases the elements intersect, but both can also be interpreted in ways separate from the other. For example, the MARC 500 field is used for notes that provide general information about the object (assuming the value does not fit into one of the specialized note fields). The DC description element is “an account of the content of the resource.” These elements could contain the same values at times (e.g., “Includes index” could be a MARC 500 note or a DC description), but either can also have values that would not be valid for the other.
- Finally, two elements can completely **exclude** each other, having no relation, e.g. MODS title and DC publisher.

Zeng & Xiao (2001) focus on the different degrees of equivalency, which is a subset of the issues encountered in crosswalking outlined by St. Pierre & LaPlant (1998). The four degrees of equivalency, as stated by Zeng & Xiao, are one-to-one, one-to-many, many-to-one, and one-to-none, each of which defines the equivalency between source and target elements, respectively.

In addition to these, St. Pierre & LaPlant identify additional crosswalking issues related to schema structure and content values. Each issue, with its description, and relationship to Zeng & Xiao's degrees of equivalency and Haslhofer & Klas' mapping expressions is presented in Table 2.2.

	Zeng & Xiao Degrees of Equivalency	St. Pierre & LaPlant Issues in Crosswalking	Haslhofer & Klas Mapping Expressions	Description
1	One to One	One to One	Equivalent	The interpretations of two elements are semantically equivalent.
2	One to Many	One to Many - identical target elements		A single source element expands into multiple occurrences of the same target element.
3		One to Many - unique target elements		A single source element expands into multiple unique elements in the target.
4	Many to One	Many to One - Combination	Include	Values of multiple source elements are mapped to a single value in the target element -- explicit rules are required to specify how the values will be appended together.
5		Many to One - Choice		Only map one source element value to a target element, with the possible consequence of data loss -- must indicate criteria for element selection.
6	One to None	Extra Source Elements	Exclude	A source element that does not map to any approrpiate element in target standard -- must specify precisely how the element value is to be added (or excluded from the mapping). Either way, there will be some data loss.
7		Mandatory Target Elements unmatched		A mandatory element in the target schema that has no corresponding mapping in the source metadata.
8			Overlap	Element definitions intersect, but are not fully equivalent, and neither element is fully included within the other.
9		Content Requirement - data type/value range		The element content must be of a specific type, i.e. specific date formatting.
10		Content Requirement - conrolled vocabulary		A target element has a controlled vocabulary, and the mapped source element has a different controlled vocabulary, or no controlled vocabulary at all.
11		Coversion combinations		Compounded crosswalking issues that present additional difficulties when mutually present.
12		Single vs Multiple Objects		Multiple object metadata concerns more than one item, while single object metadata only concerns one item per record.
13		Logical Views		A logical view enables users to see a specific set of metadata elements of the metadata standard organized in a specific way.
14		Hierarchy		The depth of the source and target schemas, which can vary greatly.

Table 2.2: Juxtaposition of Mapping Situation Delineation

2.6.2 Mapping Tools

According to Noy & Musen (2002), mapping tools "are the tools that help users find similarities and differences between source ontologies [and] either identify potential correspondences automatically or provide the environment for the users to find and define these correspondences, or both" (p. 2). In the computer science domain, research focus is surging forward with heuristics and machine-learning to help alleviate the "labor-intensive and error-prone" manual process (Madhavan, Bernstein, Domingos, & Halevy, 2002, p. 8). Yet, Madhavan, et al. acknowledge that the even the strongest mapping aids must allow for user input, and many mapping tools in the cultural heritage domain rely heavily on user interaction.

For instance, the Library of Congress provides a multitude of mapping tools. While one is a multi-program, java-based, downloadable toolkit to map between the MARC21 format and MARCXML[6], most are either mapping specifications between formats (for instance, specifying the suggested mapping from MODS to Dublin Core and vice versa) or XSLT stylesheets based on these mappings to perform the conversion of XML metadata in one format to another.[7] These tools aid in the creation of mapping specifications and completing conversions, but have little automation. In addition, they provide only blanket solutions for mappings, and are insufficient to deal with the unique nature of individual implementations of metadata schema, which can vary widely. However, by using these resources as a guide, institutions can develop specifications and conversion stylesheets tailored to their particular implementations.

In the European context, Metadata Interoperability Services (MINT) is being developed at the National Technical University of Athens, and was designed "to facilitate aggregation initiatives for cultural heritage content and metadata in Europe" (Metadata interoperability services, 2011). MINT has been used in a variety of projects, including facilitating mappings to LIDO and ESE for Europeana. It has not yet been fully developed to allow for mapping to EDM. MINT is an interactive tool, with a component that allows users to upload XML files, in order to map them to select schema. However, while it does present the elements in the particular uploaded set, semantic definitions of elements are not included within the tool – which NISO maintains is a vital factor in any mapping – nor does it provide the opportunity for users to consider the instantiated schema elements, e.g. none of the unique values are

[6] The MARC fields modeled in XML format.
[7] Available at http://www.loc.gov/standards/mods/mods-conversions.html

presented during the mapping process, examination of which would be necessary to consider the real data to be converted (Chan & Zeng, 2006).

2.7 Chapter Summary

Metadata is information about a resource, and can be administrative, structural, or descriptive. This metadata is contained within elements, and when these elements are gathered into a set, and every element is defined as to what metadata it can contain, this set is called a metadata schema. There are many and varied commonly used metadata schema throughout various domains, including Europeana's current and future schemas, ESE and EDM, respectively. These schemas can be encoded in different ways, but a very common way to encode metadata is using XML, which is a heavily utilized markup language. XML allows for easy sharing of metadata; however, even then, there are issues with interoperability.

The interoperability of metadata relates how shareable and/or reusable the metadata is outside of its original context, both with different systems and other metadata. Interoperability, which has various levels, is very difficult to achieve, as many and varied hetergeneities can occur. There are many differing methods to increase interoperability, depending on the situation at hand. One of these is the use of a uniform method for exposing XML metadata, called OAI-PMH. The most common method of achieving interoperability between two sets of metadata currently in use is the creation of mappings between metadata schemas. Metadata mappings attempt to connect the elements of one metadata schema to those of another, based on their semantic definitions and allowable values. However, this can be very difficult, as one source element does not always match one target element perfectly. There are many different situations that will be encountered when a mapping is attempted, including contraints on what values a target element can contain, significant differences in the structures of the two schema, and target elements that require a concatenation of multiple source element values or a substring of a source element value.

In order to help alleviate some of the difficulties persuent to the creation of metadata mappings, mapping tools have been created to aid in the process. These tools are extremely varied in purpose, construction, and use. There have yet to be any fully-functioning mapping tools for EDM, and the main tool in development does not incorporate within itself all of the element components that, according to NISO, must be considered in a quality mapping.

Chapter 3: Europeana

3.1 Introduction

Europeana is of central importance to this research, as the tool being analyzed would potentially be used by those isntutions that contribute to Europeana. This chapter provides an overview of what Europeana is and its purpose. In addition, the current and future metadata schemas developed for Europeana, ESE and EDM, respectively, are outlined.

3.2 Europeana

On its surface, Europeana is a centralized, web-based access portal for digital collections from cultural institutions throughout Europe, and is part of the European Digital Library Initiative, which is concerned with preservation and accessibility of European cultural heritage and scientific information (EC, 2012). The portal was launched in 2008, and is run by the Europeana Foundation. Europeana does not store digitized materials itself; rather, the site acts as

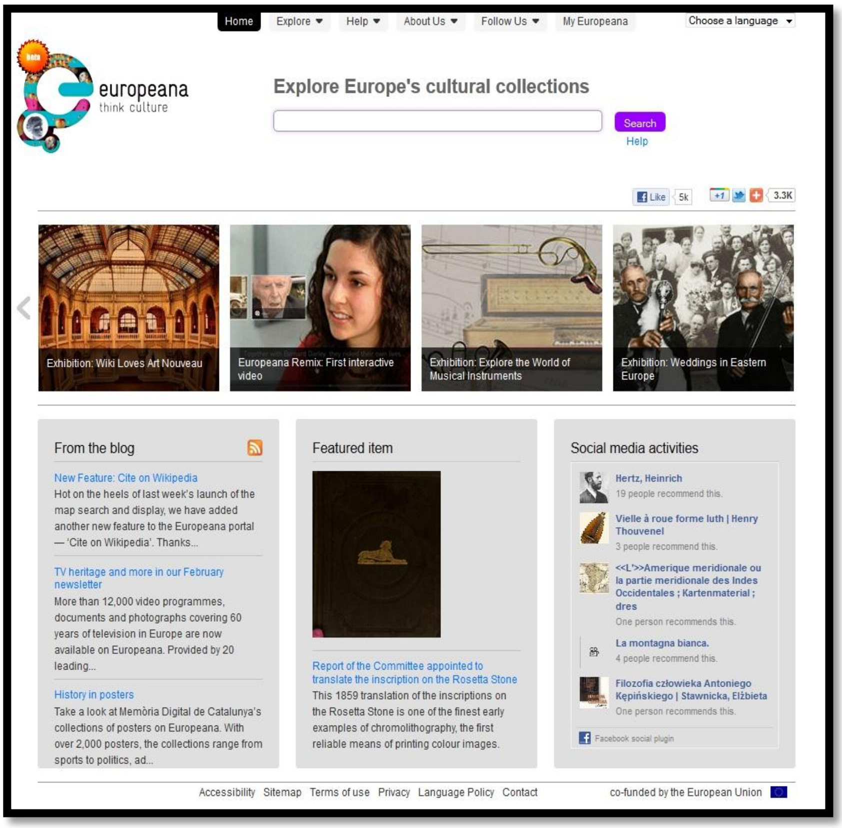

Figure 3.1: Europeana Web Portal Home Page

an entry point for object discovery, via multiple methods, including simple search, spatial and temporal visualizations, and virtual exhibitions. This eliminates the need for object exploration tools (e.g. for zoom, page turns, audio playback, etc.) and exorbitant server space, while also ensuring content providers maintain full intellectual property rights; handle preservation issues; and receive greater recognition for their content (Purday, 2009). In essence, then, Europeana is a database of cultural heritage metadata, with multiple search facilities, and was originally envisioned as “broadly open access, open source and non-exclusive” (Purday, 2009, p. 920).

When the website initially went live, Europeana contained metadata for approximately 2 million objects; now, this number is nearly 20 million, with the goal to reach 30 million objects by 2015 (EC, 2012). The strategic plan for the near future extends to more than mere content quotas (Europeana, 2011d). The plan includes four tracks: aggregation of content; facilitation of knowledge transfer throughout cultural institutions; distribution of content to users; and innovative engagement of users with content. Each track contains specific objectives. For instance, content should both be diverse in format and include representative content for all European member states (Purday, 2009).

In order to allow for the uniform distribution of and potential engagement with content, Europeana necessitates the contribution of collection metadata from participating institutions. Included in this metadata is the link to digital objects in their original context (i.e., the content providers’ digital collection website). Given the vast number of cultural institutions throughout Europe, Europeana relies largely on aggregators to collect, normalize, and expose institutional metadata. Aggregators may be national, regional, thematic, or domain-based in scope. Europeana’s preferred method of metadata retrieval is harvesting via OAI-PMH, thus, for aggregating institutions already implementing this protocol, metadata can be exposed with relative ease (Europeana Foundation, 2012).

However, there is still a major technical barrier for data providers. To ensure that metadata submitted from such widely diverse institutions can be manipulated uniformly within the centralized portal, metadata must be converted into a single, standard format using mapping techniques to transform metadata from local schemas. At Europeana’s inception, these mappings were largely performed by Europeana staff (Purday, 2009). Using the knowledge gleaned from working with the formats of the initial metadata submitted, Europeana Semantic Elements (ESE) – in multiple versions – was created and designated as the standard format, and all metadata must now be converted prior to Europeana harvesting. Currently, a new standard, called the Europeana Data Model (EDM), is being developed, and plans are being made to migrate from ESE to EDM in the near future. The specifications of these two standards are expanded in the sections below.

3.3 Europeana Semantic Elements

ESE was created specifically for Europeana, and is an extension of the Dublin Core metadata schema (Europeana, 2011). ESE includes all 15 DC elements, a portion of the DC terms element refinements, and several Europeana-defined elements. An XML schema was developed for ESE to validate submitted records and to ensure metadata follows specification rules.

ESE development was seen as necessary for the pilot launch of Europeana, in order to experience the potential of the portal (Doerr, Gradmann, Hennicke, Isaac, Meghini, & van de Sompel, 2010) . However, there are also many drawbacks to its use. First, conversion of millions of metadata records to ESE is time consuming, as many collections provided by European institutions are represented in unique implementations of metadata schemas. While generalized schema mappings can provide much guidance, the collections often must be treated individually to ensure the retention of as much metadata as possible. Still, even if mappings are performed with care, data loss is inevitable and another major drawback of ESE. For instance, records that are originally presented in semantically and structurally rich metadata formats, must lose much of this richness, as ESE is based on DC, which was created with an emphasis on simplicity and is a "lowest common denominator" approach to resource description (Purday, 2009, p. 927). In addition, as with any mapping between formats, a choice must be made between absolute and relative element mappings. Use of the former results in element values being disregarded entirely, though all mappings made will be semantically strong. Use of the latter preserves all metadata, but can result in many semantically null (or even misleading) element-value relationships in the target format.

The shortcomings of ESE that especially drove the divergence towards a new model were its inability to distinguish between the resource and its digital representation, and the inability to create semantic links, both internally between Europeana objects and externally on the web.

Two of the most obvious occurrences of a resource/representation metadata ambiguity is evident with some mappings to the *dcterms:creation* and *dc:format* elements. A 16th century painting digitized in 2005, might indicate either the date the original was painted or the date of digitization. The format element could either state the format of the original, or the file format of the digital representation. In addition, when there are multiple representations of the same real world object from different contributing institutions (as can be the case, especially with famous objects), these cannot be linked in a way to identify their shared origin. Nor can resources with the same creator, or other shared qualities, be linked in an unambiguous, semantic way. This limitation on the creation of meaningful relationships between objects detracts from the potential data reuse and manipulation which could be a substantial benefit of aggregating millions of metadata records from thousands of providers into a single portal. Acknowledging that these drawbacks of ESE were unacceptable for the long term goals of European, leaders in the Europeana community began work on the successor metadata format, EDM.

3.4 Europeana Data Model

The Europeana Data Model (EDM) is vastly different from ESE in both purpose and approach. It is also far more complex at both the conceptual and practical level. During EDM's development, – which was undertaken with the aid of technical experts spanning the cultural heritage domain – several requirements and principles emerged as critical to the model's formulation (Europeana, 2011b). These include providing a clear distinction within the model between an object, its digital representation, and its metadata; coping with multiple records for the same object; allowing for objects to be constructed of other objects, e.g., a book comprised of pages; creating metadata and vocabulary formats that are standard, but allow for specialization; handling multiple levels of abstraction; participating in an open community; allowing for rich functionality; and re-using existing models (Europeana, 2011b).

EDM is built on the idea of the Semantic Web, and is meant to support multi-directional linking between Europeana and external resources on the web, which would enable "data enrichment from a range of selected authoritative sources" (Europeana, 2011). This also led to the designation of RDF/XML as EDM's meta-model. In other words, when EDM is fully realized, all records will follow the 'triples' construct of RDF[8].

Unlike the lowest common denominator approach of ESE, EDM employs the use of a top-level ontology, which is meant to provide a comprehensive overview of the cultural heritage domain. The theoretical model re-uses several reference ontologies already in use within the Semantic Web, including SKOS, OAI-ORE, FOAF, and DC (Doerr, Gradmann, Hennicke,

[8] The Resource Description Framework (RDF) is a W3C specification for data exchange over the web (W3C, 2004). RDF is built upon the idea of triples statements for the description of relationships between resources. Each statement includes a subject, a predicate, and an object, all of which can be uniquely identified using URIs, which can be defined for both electronic and non-electronic resources (as well as concepts). With these unique identifiers, it is possible to unambiguously link statements concerning the same sources. The notion is that widespread and open use of the RDF model allows for the creation of a web of data, in which data can "be mixed, exposed, and shared" (W3C, 2004). RDF is syntax independent, though RDF/XML has been established, as well as a shorthand format for modeling called Notation3 (N3).

Isaac, Meghini, & van de Sompel, 2010, p. 4). This ontology includes core classes and contextual classes. Although all classes are concisely explained below, the prototype mapping tool will focus solely on the three core classes.

3.4.1 Core Classes

There are three core classes in EDM, differentiating the subject of the metadata provided. The most comprehensive class is *edm:ProvidedCHO*. This class contains metadata pertaining to the provided cultural heritage object (CHO), e.g. creator, title, description, etc. Similar to ESE; the properties for this class are largely comprised of DC and DC terms, with a few additional EDM elements (e.g. *edm:type*, *edm:currentLocation*, *edm:isNextInSequence*). The second class is *edm:WebResource*, which subsumes metadata associated with the digital representation of the object. The most notable properties of this class outline the intellectual property rights for the digital representation. Finally, *ore:Aggregation* is the combination of the CHO and its related web resources, and thus, properties in this class apply to the set as a whole. Aggregation metadata includes the reference for the thumbnail, the link to the digital representation in its original context, and information about the data provider. The core classes are shown in Figure 3.2.

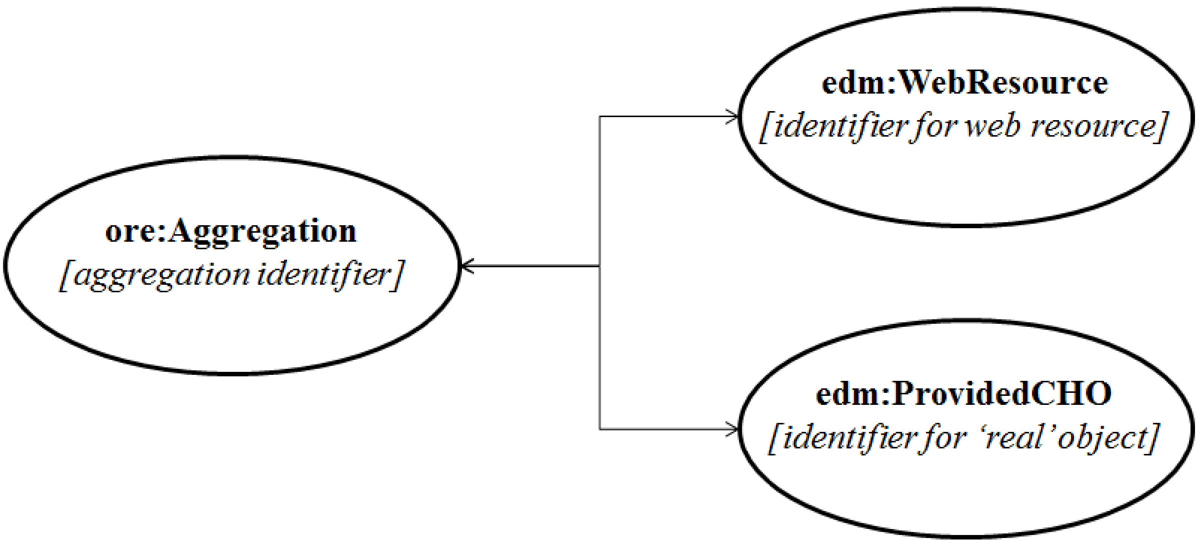

Figure 3.2: The Three Core Classes of EDM

3.4.2 Contextual Classes

Contextual classes allow for a different type of metadata separation. Often, metadata records will contain data pertaining to entities related to the object. For instance, records may include birth and death dates of the creator; information about the location or time period in which the object was created; or a reference to an authority file with a rich description of the author, place, etc. As this data does not directly describe the object, EDM outlines contextual classes to differentiate this metadata from that of the object and representation. The contextual classes are:

- *edm:Agent*: metadata pertaining to a person, or group of people
- *edm:Place*: controlled metadata pertaining to a location
- *edm:TimeSpan*: metadata pertaining to a time period
- *skos:Concept*: controlled metadata (with a URI or local identifier) pertaining to a subject/topic
- *skos:ConceptScheme*: controlled metadata (with a URI or local identifier) pertaining to a collection of concepts
- *edm:Event*: metadata pertaining to events
- *edm:PhysicalThing*: metadata pertaining to the physical item

3.4.3 Use of Proxies in EDM

The notion of proxies is an extremely important portion of the EDM framework. As such, though data providers are not required to account for the use of proxies when mapping to EDM (though it is an option in a couple of special cases), a basic understanding of how proxies are used within the data model is vital to gaining a comprehensive overview of EDM.

Any given object presented in Europeana may have multiple web representations and even multiple sources of descriptive content, nevertheless the object must still be seen as only a single, unique object (a single node in the ontological map). Yet, it is also necessary to keep each individual set of descriptive metadata about the object whole and separate. In other words, there must be a distinction between the object as described by Provider Institution 1, the object as described through any metadata enrichment performed by Europeana, and, potentially, the object as described by Provider Institution 2+ (when two or more institutions provide separate metadata for the same real-world object). To represent this distinction between each metadata sources in the ontology, Europeana follows OAI-ORE protocol and utilizes proxies. A proxy “stands in” for the resource in descriptive statements (Doerr, Gradmann, Hennicke, Isaac, Meghini, & van de Sompel, 2010; Lagoze & Van de Sompel, 2008). In the case of multiple sources of metadata, each source describes a unique proxy, allowing for all metadata to be associated with the resource, and yet still be separated. Technically, a proxy represents a resource within the context of an aggregation, which can be a collection of metadata sources pertaining to the same source or a collection of resources. The latter refers to the grouping of related resources, e.g. pages in a book. Each page is its own resource, but is also one part of a collection of sequential resources, which makes up the book. Obviously, the page as a stand-alone resource is the same as the page as part of the collection/aggregation of all pages, however, it is important (according to OAI-ORE specifications) to differentiate between the resource and the resource as part of an aggregation. Therefore, a proxy for the resource is used to reference the resource within the aggregation.

3.4.4 First Implementation

EDM was developed not as a metadata schema per se, but as a theoretical data model; therefore, there were challenges to creating a practical implementation of the model (Dekkers, Gradmann, & Molendijk, 2011, p. 6). The initial implementation, which is currently underway, will not realize the full theoretical framework of the model. In contrast with previous procedures, data providers (or the data aggregators) will provide the original (XML) metadata, along with a mapping to EDM, which allows for Europeana to make modifications to the mapping if necessary in the future without necessitating resubmission of metadata by providers (Dekkers, Gradmann, & Molendijk, 2011, p. 7).

The implementation includes seven classes, the three core classes (*edm:ProvidedCHO*, *edm:WebResource*, *ore:Aggregation*), and four contextual classes (*edm:Agent*, *edm:Place*, *edm:TimeSpan*, *skos:concept*). The classes excluded from this implementation are *skos:ConceptScheme*, *edm:Event*, and *edm:PhysicalThing*. In addition, many properties under the utilized classes will not be part of the first implementation. The active elements within the core classes of the first implementation of EDM are presented in Appendix A, as specified in the Europeana Data Model Mapping Guidelines v1.0 (Europeana, 2011c).

3.5 Chapter Summary

Given the necessity that metadata be transformed into a standard format for inclusion in Europeana, a way of easing this process would be beneficial to contributing institutions. As EDM implementation begins, tools to aid in mapping metadata to this format could be developed and examined for usability.

Chapter 4: Methodology

4.1 Introduction

This chapter explains the methodology and methods utilized in the research to investigate the research questions. This includes a short description of the purpose of the research and the qualitative methodology. The choice of research methods is detailed, and in depth descriptions of the prototyping method and the analysis method chosen, the cognitive walkthrough, are provided.

4.2 Purpose of the research

This research is exploratory in nature. The research will evaluate the usability of a proposed mapping tool designed to aid in mapping local metadata formats into EDM. As such, the findings of the study are meant to inform the potential creation of a low-barrier, browser-based mapping assistant to EDM for cultural heritage institutions, in order to analyze the functionality and usability of such a tool prior to investing significant effort into development. Should the evaluation show the design has adequate capabilities, this tool could be developed and expanded for use by cultural institutions throughout Europe that contribute content to Europeana. In addition, the analysis could inform the creation of similar low-barrier interaction tools in the future.

4.3 Methodology

This research is concerned with evaluating the usability of a system design. Usability testing is used often used in iterative system development. As such, usability testing, which "is commonly associated with systems design, development and evaluation," was considered the appropriate technique for data collection (Pickard, 2007, p. 233). Being interpretive in nature, usability testing lends itself to qualitative research, as the results are "the subjective knowledge of the researcher" and the results of usability testing rely on the interpretation of the findings by the researcher (Pickard, 2007, p. 21).

4.4 Research Methods

The creation of a fully-functioning system is a complex process with no definitive model that can be used. In fact, within the field of software engineering, the utilization of any prescriptive process model for development is debatable, as such a rigid structure can restrict the creative process required for innovation (Pressman, 2010, p. 38). Yet, without some structure, it is difficult "to achieve coordination and coherence" throughout the duration of the project, therefore a model – or a combination of several models – is often used to approach the project holistically (Pressman, 2010, p. 39). The choice of model is dependent upon the individual project, especially on the explicitness of the requirements prior to project launch.

As the project to create a mapping specification tool for EDM is exploratory, the requirements of the final system are especially vague. As Brooks (1995) states, the detailed technical requirements of a system can be the most difficult to ascertain during its conceptualization, and if these requirements are poorly specified, the system can be exceedingly flawed and difficult to fix (p. 199). Therefore, the process model chosen must provide a structure allowing for the definition of more precise system requirements. In addition, feasibility of the proposed concept must be determined. An appropriate model for this project is the use of prototyping.

4.4.1 Prototyping

When a software project is conceived with only vague objectives and feasibility is uncertain, the creation of a prototype can determine whether and how a project should move forward (Pressman, 2010, p. 43). According to Sommerville (2011), "a prototype is an initial version of a software system that is used to demonstrate concepts, try out design options, and find out more about the problem and its possible solutions" (Sommerville, 2011, p. 45). A prototype can be used to determine whether the system as proposed is feasible, and the potential areas for modification. Development of a prototype can be very beneficial when exploring potential avenues for advancement, as prototypes are generally developed in a very short amount of time. This is known as rapid prototyping (Brooks, 1995, p. 200).

There are two main approaches to the prototyping process, throwaway and evolutionary prototyping (Pressman, 2010, p. 44). Throwaway prototyping assumes that the prototype will not be utilized in the final system. The prototype is created solely for its specific purpose – whether this be demonstrable, exploratory, or problem-solving. Following the evaluation, the prototype is discarded, and system development progresses with the knowledge received from the prototype development, and without the actual software/hardware components of the prototype system. Conversely, an evolutionary prototype is meant to be the basis upon which the final system will ‘evolve;’ the prototype will be built upon to create the entire system.

Both methods have advantages and disadvantages. Throwaway prototyping allows for focus to be solely on the purpose and objectives of the prototype, which makes clarity and speed in prototype development possible. However, by discarding the prototype, the development time spent on its creation is not a part of developing the actual system. That work – while it served a purpose – is lost. Evolutionary prototyping does not have this drawback. While the prototype will serve its own unique purpose, it is made mindful of its use in the final product. This is tied to its flaw, as well, however, as future development must be taken into consideration in prototype development. Complications that can be ignored in throwaway prototyping may need to be addressed. In addition, it is likely that parts of the prototype will still need to be discarded to prevent significant flaws in fundamental specifications of the prototype from persisting into the actual system. Whether conducting throwaway or evolutionary prototyping, the prototype is not a complete system, and the prototype’s specific objectives should be made explicit, as it will not fulfill all expectations of the final system (Sommerville, 2011, p. 46).

The chosen method of prototyping greatly depends upon the context within which the prototype is being developed. Of upmost importance is a mutual understanding among all parties – prior to initiating the prototype process – as to the role the prototype will play in system development. The prototype for this research is intended to explore the usability of a design concept, and potentially outline more detailed system requirements. The final system is vague, which presents difficulties preparing the prototype for evolution. Therefore, a rapid-development, throwaway prototyping method would be the most appropriate alternative.

A widely-accepted process for prototype development, as shown in Figure 4.1, is outlined by Sommerville (2011) and aids in solidifying the purpose of prototype creation through outlining objectives and functionality.

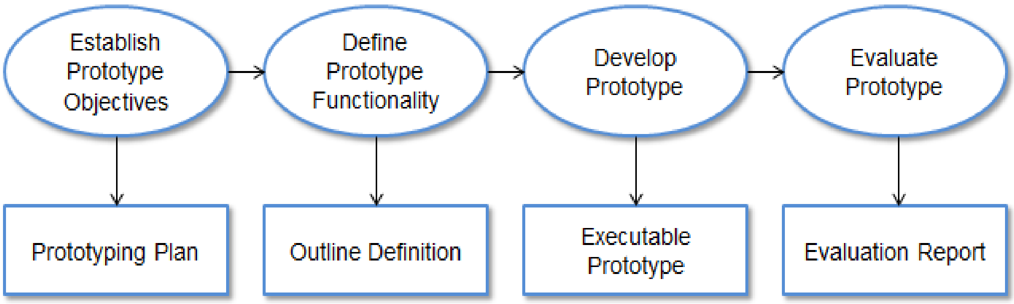

Figure 4.1: The Process of Prototype Development (Sommerville, 2011)

The development process includes four actionable steps, each of which results in a deliverable. First, prototype objectives must be established and specified in a prototyping plan. The objectives should outline what the prototyping project is meant to accomplish. Next, the prototype functionality must be defined and outlined. The outline definition should then explicate what the prototype will (and will not) be able to do. Third is the development phase, when the prototype is created. Finally, the prototype must be evaluated based on criteria derived from the initial objectives of the prototype project. The prototyping plan, inclusive of the prototype's aim and objectives, as well as a discussion of the method for prototype evaluation – derived from this prototype plan – are outlined below. Chapter 5 contains the details of the outline definition, followed by a description of the prototype development, and the executable prototype. The prototype evaluation process is provided in Chapter 6.

4.4.2 Prototyping Plan

The main objective for the creation of this prototype is to investigate whether the proposed system design's capability adequately supports typical mapping tasks in the context of EDM. The objectives of the prototype include:

- Development of a prototype of a browser-based metadata mapping assistant for Europeana content providers and aggregators, which could aid in the creation of a mapping specification from local schema in OAI-PMH conformant XML to the first implementation of the Europeana Data Model (EDM) metadata format.
- Examination of typical mapping situations in the context of mappings to the target schema EDM.
- Evaluation of the prototype's usability potential within typical mapping situations.

4.4.3 Evaluation Criteria

As previously stated, Sommerville (2011) maintains criteria for evaluation should be directly linked to the prototype objectives. The precise manner of evaluation is left open. In their discussion of ontology-mapping tools, Noy & Musen (2002) stated that there are no widely accepted criteria for the universal evaluation of mapping tools specifically, as "…mapping tools vary with respect to the precise task that they perform, the inputs on which they operate and the outputs that they produce" (pp. 2-3). As previously stated, mapping tools are extremely varied in design, intended usage, and audience. For instance, the Library of Congress's specifications and stylesheets should not be evaluated in the same manner as their interactive software for MARCXML or the European MINT interface.

As such, human-computer interaction (HCI) methods were considered for evaluation of the prototype's objective. More specifically, a form of usability testing – cognitive walkthroughs – was used in order to evaluate the system design's capability to support typical mapping situations and identify usability problems that may be encountered.

4.4.4 Cognitive Walkthroughs

Grounded in exploratory learning, a cognitive walkthrough is a "structured evaluation process" that involves pinpointing essential user tasks and analyzing in depth the design's ability to complete these tasks, typically through a series of questions applied to each step of each task individually (Mowat, 2002, p. 7; Pickard, 2007). Cognitive walkthroughs are meant to be extremely detailed in their analysis, where the validity of every actionable step of a task is brought into question in the context of the user's goals and perceived knowledge (Nielsen, 1995). The purpose of the analysis is to attempt to create "a credible story of the interaction" between the user and the interface for each task (Wharton, Rieman, Lewis, & Polson, 1993, p. 9). Then, either the task is deemed a success, or a failure with design issues extrapolated (John & Packer, 1995).

In contrast to user testing, cognitive walkthroughs are generally performed by evaluators familiar with the system – which may be developers themselves, especially in initial prototypes – who mimic user actions in order to identify problems related "to specific, key, representative user tasks" (Mowat, 2002, p. 7).

The cognitive walkthrough is a beneficial form of usability testing to perform early in the development process in order to identify and address significant flaws in the design prior to introducing users to the system, who may then be presented with a revised, more complete system and be able to address any more subtle usability flaws present. According to Sears and Hess (1999), early versions of the cognitive walkthrough are intended to investigate every step of common user tasks to identify where:

- Users would not know what they need to do next.
- Users would not be able to identify the action that would lead to a solution.
- Users would not understand how to operate a control once it is found.
- Users would not receive appropriate feedback indicating that they were on the right path for finishing the task. (p. 188)

A later, modified version of the cognitive walkthrough has similar, but divergent, investigative aims:

- **User Intent:** What the user is thinking at the beginning of an action;
- **Action Visibility:** Whether the user is able to locate the command;
- **Intent-Action Connection:** Whether the user will identify the command as the correct action; and
- **Progress:** Whether the user is able to identify progress/feedback. (Mahatody, Sagar, & Kolski, 2010; Wharton, Rieman, Lewis, & Polson, 1993)

This version of the cognitive walkthrough focuses especially on the success or failure story of each step of every task, ad consists of two phases – the preparation phase and the evaluation phase (Wharton, Rieman, Lewis, & Polson, 1993).

4.4.4.1 Preparation Phase

In the preparation phase of a cognitive walkthrough, the following must be explicated:

- A description of the users;
- A definition of the interface;
- Representative tasks to be analyzed; and
- The steps required to complete each task. (Wharton, Rieman, Lewis, & Polson, 1993)

The user description provides a profile for the intended user of the system, and the interface definition presents information about the environment of interaction between the user and the system. The tasks to be analyzed during the walkthrough must be chosen in the preparation phase. Then, for every task, the optimal sequence of steps required to complete the task is outlined. For example, a task requiring a user to save an unsaved document in Microsoft Word would start out like this:

1. Click “File” tab.
2. Click “Save” option.

This is a very simplified example, as a task in a cognitive walkthrough must be extremely explicit. For instance, the operating system must be taken into account, and the task should specify where the document should be saved and what name it should be given.

In a normal cognitive walkthrough, this is the extent of the preparation phase. To better serve the research, two additions were amended to this phase. First, as the tool is meant to be used to aid in creating a mapping specification between a source schema and Europeana’s EDM, the representative tasks were based on the mapping situations identified in the literature and previously presented in Chapter 2. In addition, as the target schema is set as EDM, it was deemed essential to consider these situations within this context specifically.

The cognitive walkthrough preparation phase in its entirety is contained within Chapter 6. This includes the user description; the interface definition; the mapping situations – and their related representative tasks; EDM considerations for each situation; and each task’s action steps.

4.4.4.2 Evaluation Phase

Once the preparation phase is completed, the cognitive walkthrough itself can be conducted. For every step of each task, a credible success or failure story was created, using the standard set of questions for this version of cognitive walkthrough:

- Will the user try to achieve the right effect?
- Will the user notice that the correct action is available?
- Will the user associate the correct action with the effect they are trying to achieve?
- If the correct action is performed, will the user see that progress is being made toward solution of their task? (Wharton, Rieman, Lewis, & Polson, 1993, p. 9)

For each of the action steps, these questions help address the four facets of the user’s cognitive process listed above. As such, usability problems can be focused within these areas and categorized as User Intent, Action Visibility, Intent-Action Connection, or Progress Issues. For every step of each representative task designated in the preparation phase, an answer to each of these four questions was provided, and detailed success or failure explanations were recorded. The analysis of the evaluation phase is in Chapter 6. Figure 4.2 presents a visualization of the entire research method process. First, the process of prototype development is begun. The *prototype objectives* are established; the *functionality* is defined; and the prototype is *developed*. Once the development is complete, the *evaluation* begins in the form of a cognitive walkthrough, which consists of two phases. In the preparation phase, the *user description*, *interface definition*, and *EDM considerations* are explicated. Then, based on *mapping situations* identified in the literature, *representative tasks* are chosen, and the individual action steps required for each task are outlined. Finally, these steps are used in the evaluation phase, where the four usability questions must be answered for every step of every task and every answer must be justified with a success/failure story.

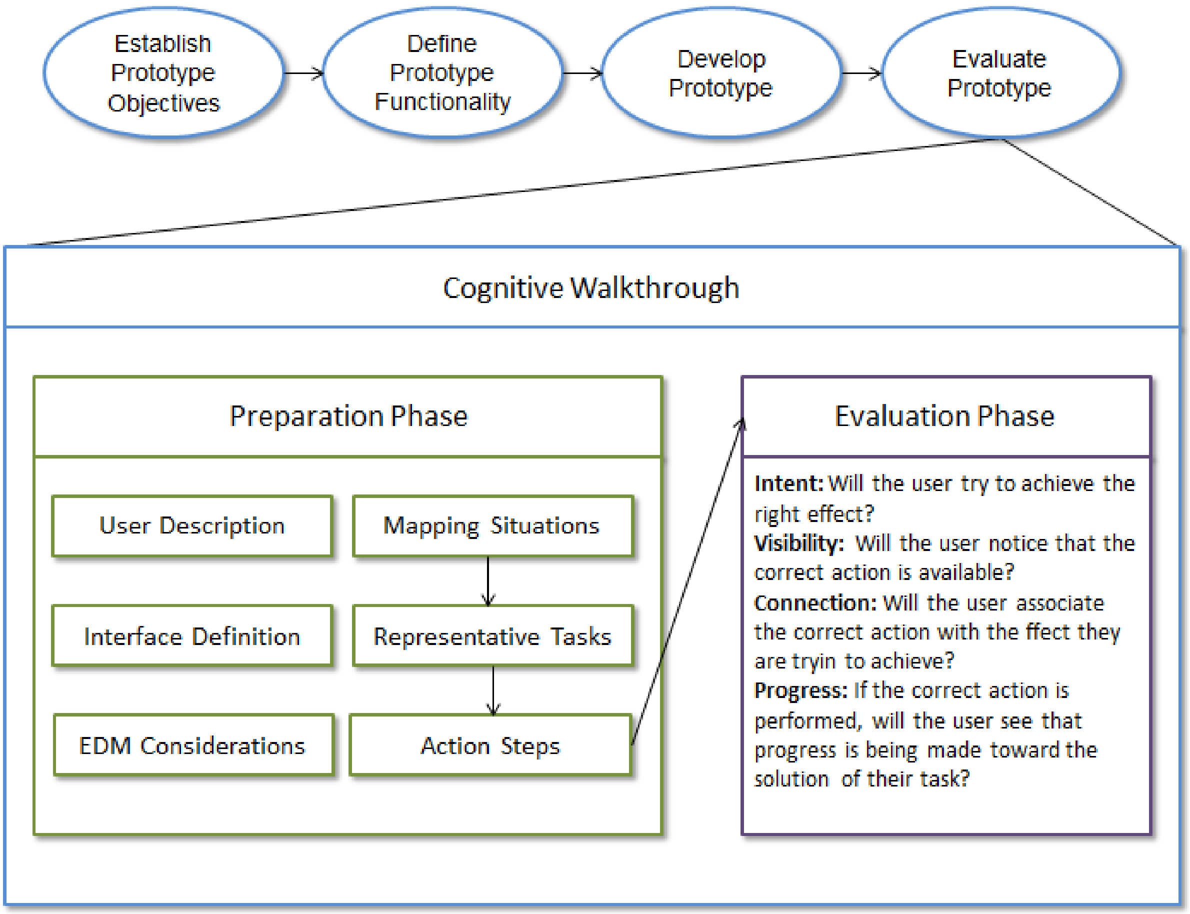

Figure 4.2: Visualization of Research Process

4.5 Chapter Summary

Through qualitative methods, the purpose of this research is to analyze the usability of a potential mapping tool design intended to aid in mapping metadata to Europeana's EDM. In order to analyze the design, a prototype was created, following the guidelines of the Process of Prototype Development. This process indicates evaluation of the prototype as the final phase, but leaves the method of evaluation open. As the main aim of the research is to evaluate the design's usability, the cognitive walkthrough, an early-stage development usability analysis method, was chosen. This method was slightly modified to include:

- Typical mapping situations as the basis for representative tasks and
- Considerations for EDM as the target schema of the mapping tool.

Chapter 5: Prototyping

5.1 Introduction

This chapter elaborates on the functionalities of the prototype, as well as the development of the prototype itself. Some of the specifics of the technical building blocks of the prototype are included; however, as the emphasis of the prototype development was on the proposed design of the user interface, the background functionality is not a focal point.

5.2 Outline Definition

The design to be analyzed through the creation of this prototype focuses on the inclusion of those element-level components that are necessary for the creation of quality mappings. As previously stated, St. Pierre & LaPlant (1998) identify four element components necessary in the creation of a metadata crosswalk. In addition, Chan & Zeng (2006) emphasize the importance of considering the “real data” present within the collection – not solely the source schema standard definitions. This provides a list of five mapping-creation necessities for elements:

- A semantic definition;
- Occurrence constraints;
- Structural constraints;
- Value constraints; and
- Actual source metadata values.

As such, the prototype design was focused on the visibility of each of these components to the user within the interface.

As the consideration of real data values requires the input of the actual metadata records, it is necessary to have the user input these records into the tool. Therefore, metadata must be uploaded to the system prior to mapping. This allows not only for the identification of real element values, but also presents all and only those elements which actually appear within the metadata records.

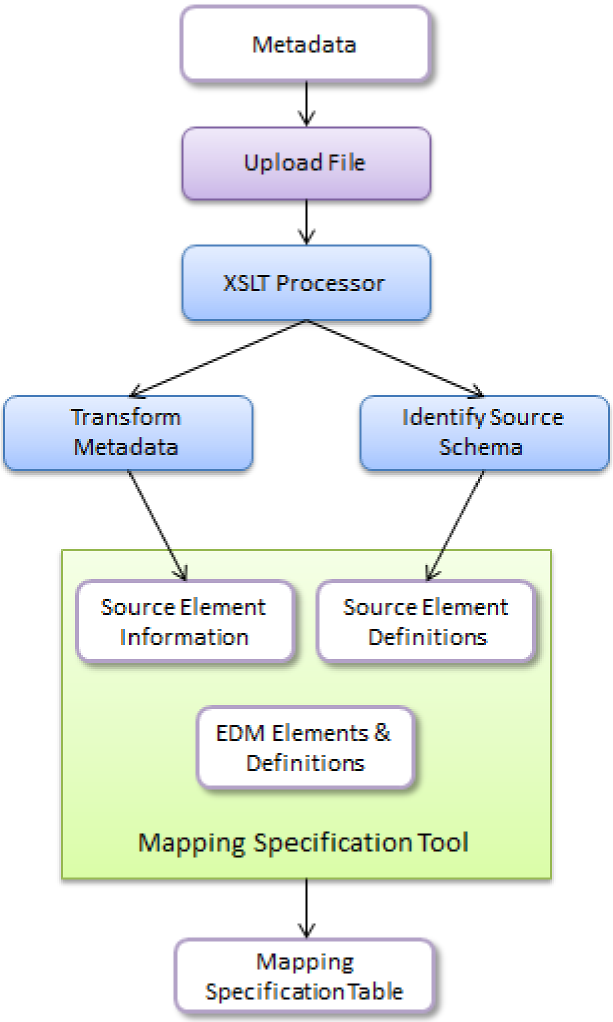

Figure 5.1: Prototype System Flow Chart

Figure 5.1 outlines the basic function of the prototype, which begins with the user uploading the metadata to be mapped to EDM. This metadata is then passed through an XSLT processor, which performs two functions. First, it transforms the metadata into the presentation of the source element information in individual rows in the mapping tool. Then, it checks whether the metadata is in a schema for which the tool has element definitions. If so, these definitions are presented alongside the source element information in the mapping tool. Each element row then includes a selection menu of all EDM elements, with the definition of the currently selected EDM element presented. The rows end in a notes section, where more complex mappings can be defined. At the bottom of the table are additional rows if needed for required EDM elements that do not map to any of the source elements. The user is meant to move essentially row by row, specifying each individual mapping. When all mappings are completed, the user clicks a button at the bottom of the mapping tool to create the specification, and all of the mappings will be arranged into a mapping specification table.

5.2.1 Constraints

As Brooks (1995) states, a prototype‘s functionality requirements are limited to the essential portions of the system the prototype is meant to demonstrate (p. 200). It is the nature of rapid prototyping to focus solely upon the objectives of the prototyping plan, especially when the prototype is created as a throwaway. As such, flaws that would not be acceptable in a finished product are inevitable. These limitations can both be explicated prior to development as constraints upon the prototype and discovered iteratively during the process. The former are the known limitations – constraints necessary when attempting to focus on a single section of a complex system. The latter is part of the purpose of the prototype – to discover the flaws and limitations of the conceptualized technical specifications. The identified constraints of this prototype include:

- A lack of scalability, i.e. the size of the XML document to be uploaded is limited, due to server limitations.
- A select number of source metadata schemas (MODS, DC) with element descriptions available, due to time limitations.
- Source schemas must be tag-centric in their XML iterations (as opposed to attribute-centric, e.g. MARC XML), as the XSLT in use is based upon this assumption.

5.3 Prototype Development

Prototype development began with the establishment of an Apache server to simulate the prototype tool as a website. This allowed the prototype to actually ‘accept’ any uploaded files to be displayed in the mapping tool page. Figure 5.2, shows the simple page which was created to allow for prototype upload.

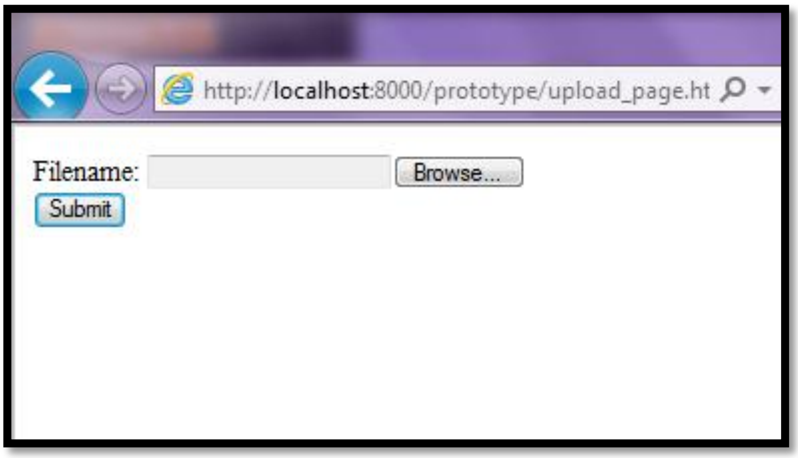

Figure 5.2: Prototype XML File Upload Page

Here, the user would browse for the appropriate XML metadata file. For the prototype, the XML file must be formatted as OAI-PMH compliant metadata, a sample of which is displayed below.

```
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:marc="http://www.loc.gov/MARC21/slim"
xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
   <responseDate>2012-03-01T08:44:20Z</responseDate>
   <request verb="ListRecords" metadataPrefix="mods"
set="coll">http://memory.loc.gov/cgi-bin/oai2_0</request>
   <ListRecords>
      <record>
         <header>

<identifier>oai:lcoa1.loc.gov:lccn/2002556033</identifier>
            <datestamp>2008-05-01T15:49:49Z</datestamp>
            <setSpec>coll</setSpec></header>
         <metadata><mods xmlns="http://www.loc.gov/mods/v3"
xmlns:xlink="http://www.w3.org/1999/xlink"
xsi:schemaLocation="http://www.loc.gov/mods/v3
http://www.loc.gov/standards/mods/v3/mods-3-1.xsd">
               <titleInfo>
                  <title>History of the American West,
1860-1920</title>
```

Metadata formatted as OAI-PMH results allowed for a simplification of the prototype – as all of the input metadata would have the same container structure, manipulating the metadata via XSL was more feasible. When the file is submitted, it would then be processed by the following PHP.

```
<?php
 copy ($_FILES['Image']['tmp_name'],
$_FILES['Image']['name']) or die ('Could not upload');
 $node = $_FILES['Image']['name'];
 # LOAD XML FILE
 $XML = new DOMDocument();
 $XML->load($node);
 # START XSLT
 $xslt = new XSLTProcessor();
 $XSL = new DOMDocument();
 $XSL->load( 'Transform3.xsl', LIBXML_NOCDATA);
 $xslt->importStylesheet( $XSL );
 #PRINT
  print $xslt->transformToXML( $XML );
?>
```

The first two lines of this code copy the file to the server. Next, the uploaded file is identified as an XML document. Then, the PHP XSLT extension is used to process the uploaded XML document through the already created XSL document on the server (Transform3.xsl).[9] Finally, the resultant document is displayed in the browser as the mapping tool.

In addition, the metadata schema in use is identified through the OAI element *metadataPrefix*. This is used to include the element definition for each element within the schema present in the source metadata. The element definitions were stored in a separate XML file, which was referenced by the XSL during the transformation process. For the prototype, only MODS and DC element definitions were included in this document.

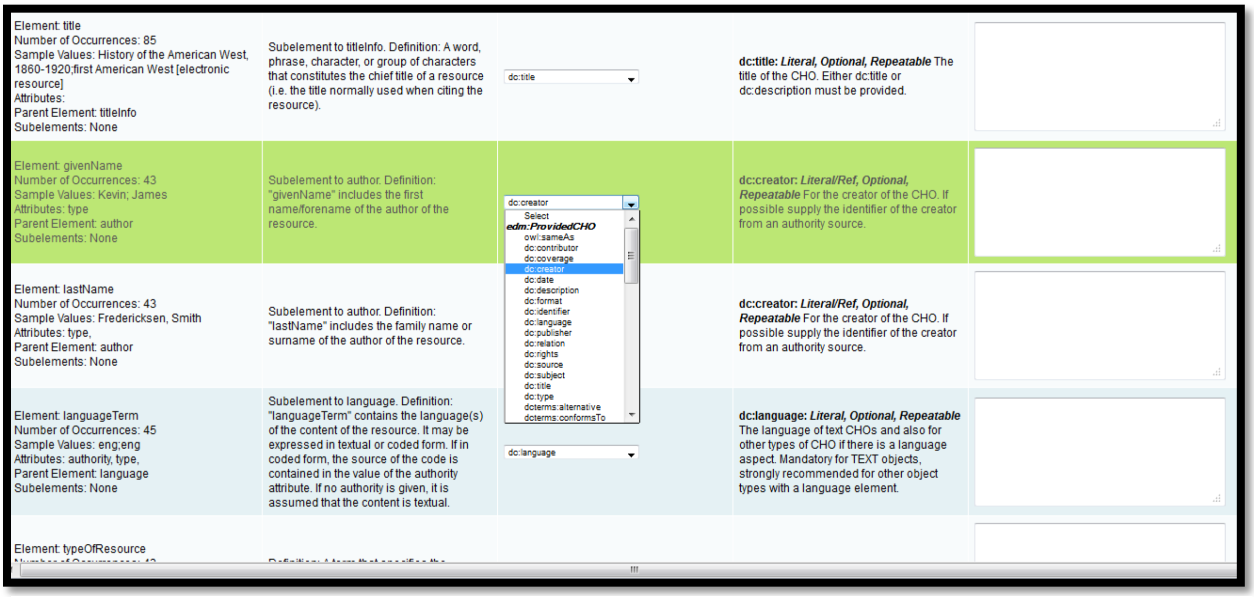

Figure 5.3: Prototype Tool Display

As shown in Figures 5.3 and 5.4, the mapping assistant page has a five-column table-based design. Every element within the source metadata file has an individual row. In the first cell of each row, the following information extrapolated from the metadata is presented:

- The element name,
- Number of occurrences in the source document,
- Attributes,
- Sample values,
- Parent element, and
- Child elements.

[9] Transform3.xsl, in its entirety, can be found in Appendix B.

The second column includes the imported element components (definition and constraints).[10] The third column has a dropdown list of all EDM core class elements used in the first implementation.[11] Using Javascript,[12] the fourth column displays the element components for the selected EDM element. The final column includes a free text notes section. For each element, the user chooses the appropriate EDM element, using the definitions as a guide, and includes any specifics in the notes section. Additional rows at the bottom of the table are available for required EDM elements that may not be present in the source metadata (for example, the *edm:dataProvider* element). In the eventual fully-functioning system, when all mappings have been made, the user would be able to click the "Make Mapping Specification" button to produce a printable table.

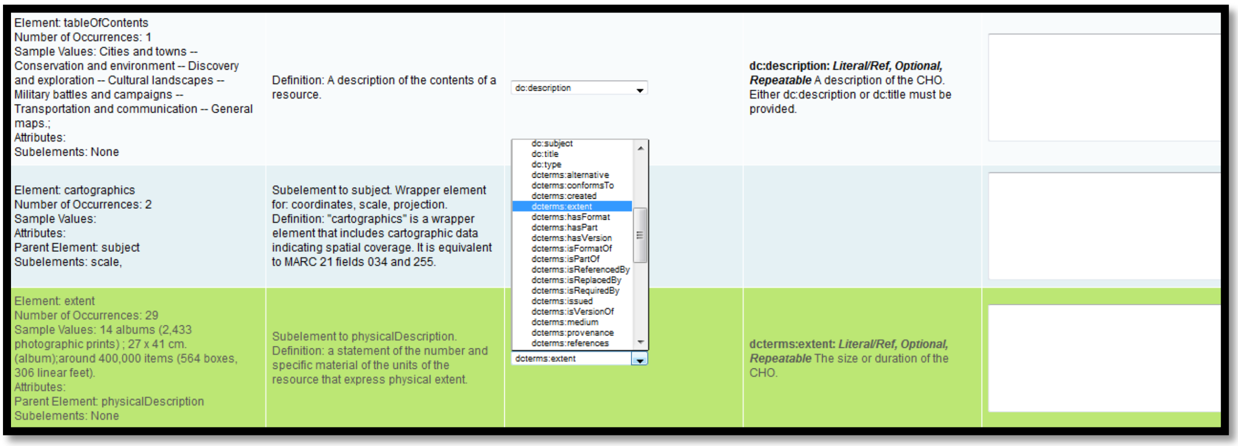

Figure 5.4: Prototype Tool Display

5.4 Chapter Summary

The prototype was developed to provide the opportunity to analyze the usability of a possible mapping tool design. The prototype was capable of uploading an XML file of collection metadata in OAI-PMH format; presenting this metadata by the elements contained in the document; importing element definitions for MODS and DC schemas; and providing an environment to map the metadata elements to the elements of EDM.

Once the prototype development stage was complete, the prototype evaluation stage could begin. In the case of this research, the preparation phase of the cognitive walkthrough was begun.

[10] An excerpt from the XML document containing source element definitions is in Appendix C.
[11] Appendix D contains the EDM dropdown menu file.
[12] The Javascript file containing EDM element definitions and constraints is in Appendix E.

Chapter 6: Analysis

6.1 Introduction

The cognitive walkthrough consists of two phases. During the first phase – the preparation phase, a description of the anticipated system user and a definition of the system were explicated. Then, mapping situations previously outlined in the literature were used as guidelines for the creation of representative tasks likely to be undertaken by the user. The sequence of steps required to complete each task was sketched, and any considerations specific to the mapping situation in the context of mapping to EDM were identified.

The second phase, the evaluation phase, consisted of a "walkthrough" of each of the tasks. For every task, each step was examined individually, analyzing the usability issues that could be encountered dealing with user intention, action visibility, the user's connection of their intention to the correct action, and perceived progress. These usability facets were judged by answering a standard set of four questions used in a version of the cognitive walkthrough.

6.2 Preparation Phase

Table 6.1 outlines the user description, which provides a profile of the anticipated user of the mapping tool. This is important to consider for the duration of the cognitive walkthrough. For instance, the user description states that the user "will likely have a good understanding of…using HTML forms." Therefore, during the cognitive walkthrough, assumptions can be made that the user would recognized a dropdown menu, a text box, etc.

User Description
The intended users for this tool are workers at cultural instutitions throughout Europe that contribute metadata to Europeana. They will likely have a good understanding of metadata, and be familiar with working in a browser and using HTML forms. The user would have at least a basic understanding of EDM and the source metadata schema, as well as the creation of metadata mapping specifications. The user may not know details concerning the instantiated metadata they upload into the tool, e.g. which element fields were utilized and how they were utilized.

Table 6.1: Description of Anticipated Users of Mapping Tool

A short interface definition is presented in Table 6.2. For a more detailed explanation of the interface, refer to the Outline Definition in Chapter 5.

Interface Definition
The interface is browser-based. The mapping specification tool is an HTML form set up in a tabular structure - utilizing dropdown menus, radial buttons, and free texts sections to allow for the creation of a mapping specification.

Table 6.2: Definition of Interface for Mapping Tool

The representative tasks to be evaluated were chosen to examine the tool's ability to handle the various mapping situations that may be encountered when creating a mapping specification, especially as they relate to the EDM target schema. These situations were based upon those outlined previously in the literature, combining situation categorizations presented by Zeng & Xiao (2001), St. Pierre & LaPlant (1998), and Haslhofer & Klas (2010). The fourteen unique situations presented were:

1. One-to-One
2. One-to-Many: Identical Target Elements
3. One-to-Many: Unique Target Elements
4. Many-to-One: Combination
5. Many-to-One: Choice
6. One-to-None
7. Mandatory Target Elements Unmatched
8. Overlap
9. Conversion Combinations
10. Content Requirement – Data Type/Value Range
11. Content Requirement – Controlled Vocabulary
12. Single vs. Multiple Views
13. Logical Views
14. Hierarchy

Of these situations, the first eleven center on element situations, and were used as the basis of the representative tasks. The final three – Single vs. Multiple Views, Logical Views, and Hierarchy – are schema level situations and were not examined with this cognitive walkthrough, as it would be difficult to represent these types of situations in a single, straightforward task, as necessary for the cognitive walkthrough.

For each of the eleven element-level situations, a task was created to represent the mapping situation. There was much subjectivity in the creation of these tasks, as every mapping situation could result in limitless potential tasks. Some of the tasks were very exact and concrete, concerning a mapping of a real world element(s) to that of one or more EDM element(s). For example, the task for the first situation used the MODS element *title*:

One-to-one. Being semantically equivalent, mods:title should be mapped to dc:title. Make the mapping tool display this specification.

Other tasks were more hypothetical in nature, using representative elements instead of a specific schema. For example, the task for the Overlap situation does not specify a schema for the source elements. Instead, a general source element, *info*, with a definition that only partially matches that of *edm:description*, must be mapped. In addition, the Extra Source Elements situation task only specifies that the source element has no mapping match in EDM. Table 6.3 shows all of the mapping situations and their representative tasks.

Once the tasks were created, every task was expanded into an action narrative. That is, the user actions required to fully complete the task were outlined in detail. The action narrative was then separated into the individual sequential steps of the task, and each of these steps was analyzed through the questions of the cognitive walkthrough. The task steps can be found in the cognitive walkthrough, which is in Appendix F.

Mapping Situation	Representative Task
One to One	Being semantically equivalent, mods:title should be mapped to dc:title. Make the mapping tool display this specification.
One to Many - identical target elements	A single source element subjects includes all subjects for the record, separated by semi-colons, which should be unique occurrences of dc:subject in EDM. Specify this with the mapping tool.
One to Many - unique target elements	In a single field, the main author is listed first, then the word "contributors:", followed by semi-colon separated contributors to the work. Specify that the main author must be in dc:creator, while all others should be in dc:contributor.
Many to One - Combination	givenName and lastName should be combined into a single edm:creator field. Specify this with the mapping tool.
Many to One - Choice	The source metadata has two location elements for the source object - the first with the building and the second with the coordinates, but EDM only allows one edm:currentLocation. Specify a mapping that only includes the coordinates in edm:currentLocation, and includes the building in a dc:description field.
Extra Source Elements	The source metadata has an element that has no exact or approximate match in the EDM elements.
Mandatory Target Elements unmatched	There is no source element matching the required edm:DataProvider element. Specify that the data provider is the the same for all of the metadata records.
Overlap	Map an info field in the source metadata to EDM, though there is no absolute match in the target schema.
Content Requirement - data type/value range	The values of the source element date do not conform to the recommended date formatting for dc:date.
Content Requirement - controlled vocabulary	edm:type has 5 allowable values. Specify that the source metadata is a collection of all images.
Conversion combinations	(many-to-one combination & content requirement - data type) The source elements day, month, and year, must be mapped to dc:date, which has a recommended value format.
Single vs Multiple Objects	Not evaluated
Logical Views	Not evaluated
Hierarchy	Not evaluated

Table 6.3: Mapping Situations and Representative Tasks Utilized in Cognitive Walkthrough

6.2.1 EDM Considerations

The mapping situations compiled from the literature apply to mappings in general – when two schemas must be mapped to each (or metadata in one schema must be mapped into another). In the context of this mapping tool, however, metadata is always being mapped to the target schema EDM. Therefore, it is important to consider how the fixed EDM schema affects each situation. For some situations, the EDM context provides no insight particular insight; for others, however, this context can be informative.

One-to-One

There are no EDM-specific considerations for one-to-one matches, except that any source metadata schema based on Dublin Core will likely have many one-to-one relationships, as EDM is also based on Dublin Core.

One-to-Many – Identical Target Elements

There are no EDM-specific considerations for one-to-many identical target element situations. However, it should be noted that most EDM fields are repeatable, which may result in mapping to multiple occurrences of the same target element from a single source element with complex content values (if, for instance, the source metadata has all the record's subjects or authors in a single field).

One-to-Many – Unique Target Elements

There are no EDM-specific considerations for one-to-many situations with unique target elements, though it is certainly possible such situations may occur.

Many-to-One – Combination

There are no EDM-specific considerations for many-to-one combination situations. though certainly when mapping from metadata in a schema like MODS, which separates title and subtitle, for example, this situation will certainly occur.

Many-to-One – Choice

It is unlikely that the many-to-one choice situation would happen very often when mapping to EDM. EDM tends towards flexibility when binding element occurrences in an individual record. There are a limited number of elements that may only occur once in a record, and presumably these are the elements that would lead to a many-to-one choice situation. These elements are generally unambiguous. For example, *edm:currentLocation* is for the physical location of the cultural heritage object portrayed by the digital representation. The object can only be in a single location. Where this situation may occur, however, is when there are two separate source elements presenting this information in different ways. To take the current location example, there may be an element given a building, city, country, etc. while another source element gives the exact coordinates of the object. As the EDM element is not repeatable, only one of these elements may be mapped to *edm:currentLocation*, which results in a many-to-one choice situation (or potentially a many-to-one combination situation, depending on the decision of the mapper).

Extra Source Elements

The full implementation of EDM is intended to make use of RDF Linked Data, in that any source metadata that is specified using linked data elements will be able to 'carry over' these elements, even if they don't map well to any EDM elements. They will be linked over the Web of Data. This would allow for virtually no data loss for the metadata stored by Europeana, making the extra source elements situation less of an issue. However, for the purposes of display and use within the Europeana portal, metadata will still be mapped to the EDM elements, and again there may be some source elements that do not adequately map to any available EDM elements.

Mandatory Target Elements Unmatched

There are only five fully-mandatory elements in the core classes of the first implementation of EDM. *edm:type*, which falls in the *edm:ProvidedCHO* class, is required. The remaining mandatory elements (*edm:aggregatedCHO*, *edm:dataProvider*, *edm:provider*, and *edm:rights*) all fall under the *ore:Aggregation* class. These would generally be specified at collection level, and would not necessarily be dependent on the value of any particular source element. In this way they are removed from the element-to-element mapping. Additionally, there are three sets of conditionally mandatory elements.

- In the *ore:Aggregation* class, either *edm:isShownAt* or *edm:isShownBy* must be present. As Europeana is meant for metadata of digital representations, it is logical that there is an online accessible location where the digital representation can be seen in its original context. However, it is possible that this data is stored separately from the descriptive metadata. If this is the case, a more sophisticated solution must be devised to converge these two metadata sources.
- In the *edm:ProvidedCHO* class, at least one of the following four elements must be present: *dc:subject*, *dc:coverage*, *dc:type*, *dcterms:spatial*.
- dc:language must be provided for text objects.

Overlap

There are no specific EDM considerations for overlap situations, which occur when a source element definition partially corresponds to that of a target element, but the definitions diverge in some (or many) instances. Given the diversity of the cultural heritage domains converging in Europeana, this situation is very plausible.

Content Requirement – Data Type/Value Range

With the heterogeneity of metadata contributors to Europeana, requirements must be extremely lax in regards to the content values of elements, as reformatting of value strings can be exceedingly complex. As such, EDM has only two basic value requirements for its elements: literal and reference. Many of the elements can have *either* a literal value or a reference value (either a URI or an internal reference number). Some are only allowed literal values, others only reference values. These are the only real value range requirements for elements in the EDM core classes. As such, the effect of this situation would likely be minimal.

The number of core class elements that only allow literal values or reference values is relatively low (7 and 9, respectively, out of 52 total elements). In addition, the specified data types themselves are extremely broad. And, especially in the case of the elements allowing only literal values, the type is more concerned with how the content is handled within the EDM ontology.

```
IDLE 2.6.6
>>> m = 17
>>> n = '17'
>>> print m
17
>>> print n
17
>>> type(m)
<type 'int'>
>>> type(n)
<type 'str'>
>>> m + 1
18
>>> n + 1

Traceback (most recent call last):
  File "<pyshell#7>", line 1, in <module>
    n + 1
TypeError: cannot concatenate 'str' and 'int' objects
>>> n + '1'
'171'
>>>
```

Figure 6.1: Example of Programming Language Value Types

A comparison could be made in the typing of variables in programming languages. Figure 6.1 exemplifies this. Where *m*=17 and *n*='17', the programming language python will return the same value for each when instructed to print. However, *m* is type 'integer' and *n* is type 'string'. Since *m* is an integer, it can be added to 1 to return the number 18. *n*, however, cannot be added in such a way, and attempting returns an error (though, the value can be 'added' to other strings, like '1', in a concatenation). Similarly, though the content of an element could be considered referential, if it is mapped to an EDM element that allows only literal values, it will be considered a literal within the model.

Naturally, elements that require reference values cannot be treated in the same way, just as a literal string could not be an integer number. However, for most of the elements in question no literal value would be acceptable under the element definition. These are elements requiring reference values:

- *edm:rights* requires its value to be taken directly from the Europeana Rights Guidelines.
- *edm:isShownBy*, *edm:isShownAt*, *edm:Object*, and *edm:hasView* all refer in various ways to the web location of the digital representation of the cultural heritage object in its original context.
- *owl:sameAs* is meant specifically for the URI of an object if it is already a part of linked data.
- *edm:isNextInSequence* requires the unique identification of the item preceding this one (for instance, the preceding page in a book).
- *edm:aggregatedCHO* requires the unique identification of the cultural heritage object itself.

The elements that deal with date values (*dc:date*, *dcterms:issued*, *dcterms:created*) do have recommendations for formatting of dates, though they are not requirements. Though this is not a *required* format, as the situation name proclaims, it is used as the representative task for demonstrable purposes.

Content Requirement – Controlled Vocabulary

There are two elements in EDM core classes that have specified, required controlled vocabularies. The first, *edm:type*, only has five allowable values: IMAGE, TEXT, VIDEO, SOUND, and 3D. This allows all of the records within Europeana to be separated based on these criteria. The second element is *edm:rights*, which holds the rights statement for the digital representation of the cultural heritage object. The value must be taken from a list provided by Europeana.

EDM as a theoretical model is meant to incorporate the controlled vocabularies used in the source metadata, though this would not result in a problem mapping situation, as the source elements would merely carry over the controlled vocabulary they already have in use.

Conversion Combinations

There are no EDM-specific considerations for conversion combination situations, which occur whenever two or more issues occur within the same element mapping. These situations are obviously extremely varied and come in a plethora of forms. Therefore, the task for this situation is not necessarily very representative of the situation group as a whole, merely one potential occurrence and its outcome.

Single vs. Multiple Objects, Logical Views, and Hierarchy

These three situations focus on record level heterogeneities, and are excluded from the cognitive walkthrough.

6.3 Evaluation Phase

A cognitive walkthrough was performed on tasks representing the eleven element mapping situations identified through the literature review. Then, each of these tasks was extrapolated into an action narrative, which included every step required to complete the task. The eleven tasks were expanded into a total of 91 action steps. The number of steps for each task is presented in Table 6.4. Each action step was analyzed through the four cognitive walkthrough usability questions:

Task	**No. of Steps**
One-to-One	6
Content Requirement - Controlled Vocab	8
Unmatched Mandatory Target Elements	7
One-to-Many Identical Target Elements	9
Many-to-One Combination	17
Many-to-One Choice	9
One-to-Many Unique Target Elements	10
Overlap	6
Unmatched Source Elements	5
Content Requirement - data type/value range	7
Conversion Combinations	7
Total	91

Table 6.4: Number of Steps to Complete Each Task

- **Intent:** Will the user try to achieve the right effect?
- **Visibility:** Will the user notice that the correct action is available?
- **Connection:** Will the user associate the correct action with the effect they are trying to achieve?
- **Progress:** If the correct action is performed, will the user see that progress is being made toward solution of their task? (Wharton, Rieman, Lewis, & Polson, 1993, p. 9)

Task	Will the user associate the correct action with the effect they are trying to achieve?	Will the user see that progress is being made toward the solution of their task?
Compares to source element definition. Sees that this is appropriate for the author.	Yes. Once the definition appears, the user will again want to compare it to the definition and information they have available for the source element.	Yes. As the user compares the source element information and definition to that of the target element, they will see that dc:creator is the proper field for the source metadata suthor element.
Cannot choose two target elements for a single source element. Instead, chooses dc:creator, and writes in Note field: "dc:creator for content before 'contributor:'. Create separate dc:contributor elements for each semi-colon separated name after 'contributor:'."	No. The user may not realize that the way they are meant to specify that the values in the source element should be separated is by writing an explicity note in the Note field. They may expect something more concrete -- something that is machine readable, for instance. If they do realize this is where they are meant to specify such instructions, they will likely type in an adequately specific note to inform the mapping.	No. While the note will extrapolate the mapping, the user may find this solution wanting.

Figure 6.2: Excerpt from Cognitive Walkthrough

Task	Will the user try to achieve the right effect?	Will the user notice that the correct action is available?
One-to-One		
The user must first identify the row for the mods:title element	Yes. Given that the user is attempting to make a mapping between their source metadata and EDM, they will anticipate making element-by-element mappings, and will therefore focus on one element at a time. This example has them focusing on the mods:title element.	Yes. The righthand columns all start out identical in each row, the two columns on the left are unique in each row. A user used to using tables will look to the top row for column headings and read columns left to right. Looking at the left most column, the user will be able to identify that each row represents a separate element, and be able to locate the row for the mods:title element
Read the information about the source metadata and the element description.	Yes. After identifying the correct row for the element they are trying to map, the user may want to know more precise information about the use of the element field, and would likely want to consult the element definition prior to making a mapping decision.	Yes. The element description is directly to the right of the element name making it very visible for the user. It follows a logical left-to-right eye movement.
Click the dropdown menu	Yes. The user is making a mapping between two metadata schema, and will know that they must make element mappings. After looking over the information for a source element, they will next want to look at the potential target elements available to find the best match.	Yes. The dropdown menu is direclty to the right of the element definition, and as the user continues to follow the left-to-right eye motion, they will see the dropdown menu for the row and along with its default value ("Select an EDM element") and realize this is how they proceed with the mapping. The column heading will also give an indication that this is where the EDM element is chosen.
Reading through the available elements, dc:title is chosen for its syntactic similarity to the source element name	Yes. For an element like mods:title, which is relatively straight forward, the user is likely to scan through the available target elements to find one that looks like it would contain a title.	Yes. After clicking into the dropdown menu, the user will infer that they are meant to make a choice of one of the options below. Looking down the list, the user will see dc:title as an option.

Figure 6.3: Excerpt from Cognitive Walkthrough

As these four questions were asked of each of the 91 steps, the cognitive walkthrough resulted in 364 answers and corresponding justifications. Figures 6.2 and 6.3 show small excerpts of the walkthrough responses. 26 (28.57%) of the 91 action steps resulted in a failure story for at least one of the four usability questions, and there was a total of 43 failure stories in the cognitive walkthrough. These failure stories indicate usability issues, which fall into different categories (intent, visibility, connection, or progress) depending on the question that resulted in the failure. Common failures occurring in multiple tasks were coded into the same issue. Table 6.5 shows the differentiation of these issues into these four categories cross-referenced with the mapping situations in which the issues occurred. For example, the cognitive walkthrough identified 16 total connection failures, resulting in five unique connection usability issues. The Content Requirement-Data Value task accounted for one of these: ‘Confusion using Note field to outline mapping specifications.’

Potential Issues Identified through Cognitive Walkthrough	**11**	**M1-1**	**M1-2**	**1M-1**	**1M-2**	**1N**	**N1**	**O**	**CRD**	**CRC**	**CC**	**Total**
Unlikely to navigate as expected between dispersed information		X		X								2
Confusion from switching between multiple rows			XX								X	3
May not reference source metadata information as expected		X	X	X	X							4
May decide not to explain relative mappings fully								X				1
May not ensure mandatory target elements are fulfilled							X			X		2
Uncertain how to proceed with 2 targets for one source element					X							1
Confusion with compound source element					X							1
Total Intent Issues	**0**	**2**	**3**	**2**	**3**	**0**	**1**	**1**	**0**	**1**	**1**	**14**
Unlikely to identify specific information in first cell in element row		X							X			2
Difficulty identifying potential EDM elements when source info provides no guidance						X						1
Unlikely to identify details about sample values		X										1
Initial uncertainty of EDM definition location						X						1
Total Visibility Issues	**0**	**2**	**0**	**0**	**0**	**2**	**0**	**0**	**1**	**0**	**0**	**5**
Confusion using Note field to outline mapping specifications		X	X	X	X				X		X	6
Uncertain whether specification should be copied if multiple source elements are combined into one element			X								X	2
Uncertainty when source element has no syntactic (and possibly semantic) match in EDM list						X		X				2
Difficulty connecting certain pieces of information to draw conclusions		X		X	X							3
May not draw conclusions from sample values		X	X	X								3
Total Connection Issues	**0**	**3**	**3**	**3**	**2**	**1**	**0**	**1**	**1**	**0**	**2**	**16**
Not certain whether mapping specification is complete for element due to reliance on note field			X		X				X		X	4
Uncertain about progress when jumping between rows			XX								X	3
Uncertain about progress when source element has no match						X						1
Total Progress Issues	**0**	**0**	**3**	**0**	**1**	**1**	**0**	**0**	**1**	**0**	**2**	**8**
Total	**0**	**7**	**9**	**5**	**6**	**4**	**1**	**2**	**3**	**1**	**5**	**43**

11: One-to-One; **M1-1:** Many-to-One-Choice; **M1-2:** Many-to-One-Combination; **1M-1:** One-to-Many-Identical; **1M-2:** One-to-Many-Unique; **1N:** Unmatched Source Element; **N1:** Unmatched Mandatory Target Element; **O:** Overlap; **CRD:** Content Requirement-Data Value; **CRC:** Content Requirement-Controlled Vocabulary; **CC:** Conversion Combination

Table 6.5: Presentation of Issues Identified through Cognitive Walkthrough Analysis

More specific details concerning the issues encountered in each of the usability categories – user intent, action visibility, action-intent connection, and response – are presented in the sections below.

6.3.1 User Intent Issues

A user intent issue is identified in the cognitive walkthrough when the action step would not be part of the natural progression through the task for the user. In answering the question "*will the user try to achieve the right effect?"* for the action step, the cognitive walkthrough examines whether or not the user will ascertain that this action is what they are supposed to do following the completion of the preceding step (or, in the case of the first step, whether it would be their initial action to attempt task completion).

There were 14 user intent issues identified by the cognitive walkthrough. A major intent issue that occurred multiple times and in several tasks, was how the users would interact with the information provided. The cognitive walkthrough showed that optimal task completion occasionally expected the user to navigate the information throughout the mapping tool in potentially unlikely ways. The task anticipated that users would reference the source element information provided through the metadata when there is little indication to do so. The identification of the action was contingent upon the user recognizing certain segments of the element information extrapolated from the source metadata and drawing conclusions based on this information – which would then affect their mapping decisions.

For example, in one-to-many situations, the user must realize that the mapping should split a single source element into multiple target elements (either different elements or multiple occurrences of the same element). In order to come to this conclusion, they may have to identify and carefully examine the source element sample values presented in the mapping tool; compare this information with one or more EDM element definition(s); and determine that the source element value should be separated into multiple target elements. One of the many-to-one tasks consisted of a single source element containing all of the subjects for the record, separated within the field by semicolons. In EDM, each subject should be in separate *dc:subject* fields. As can be seen in Figure 6.4, looking at the sample values in the mapping tool, it would be difficult for the user to even realize that the subjects are in the same field in the source metadata, unless they were specifically focused on this. And if the user does not notice the compound source element, they will not be able to create a specification that accounts for the situation.

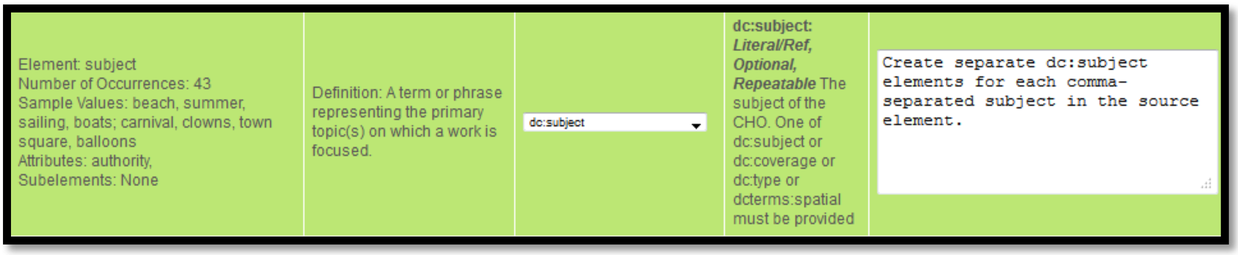

Figure 6.4: User Intent Issue Example: One-to-Many – Identical

There could also be instances where the user would have a difficult time drawing conclusions even after carefully analyzing all the information provided in the mapping tool. For instance, if the user is mapping an element to an EDM element that is non-repeatable, it could be difficult to impossible to deduce whether the source element is repeated in any of the metadata records from the information in the mapping tool. The sample values and the number of total occurrences of the element (compared to the total number of records) could provide some indication, however, this is certainly not guaranteed. While in Figure 6.5 it is possible to see that the element *location* occurs twice as many times as there are number of records, it is not immediately apparent.

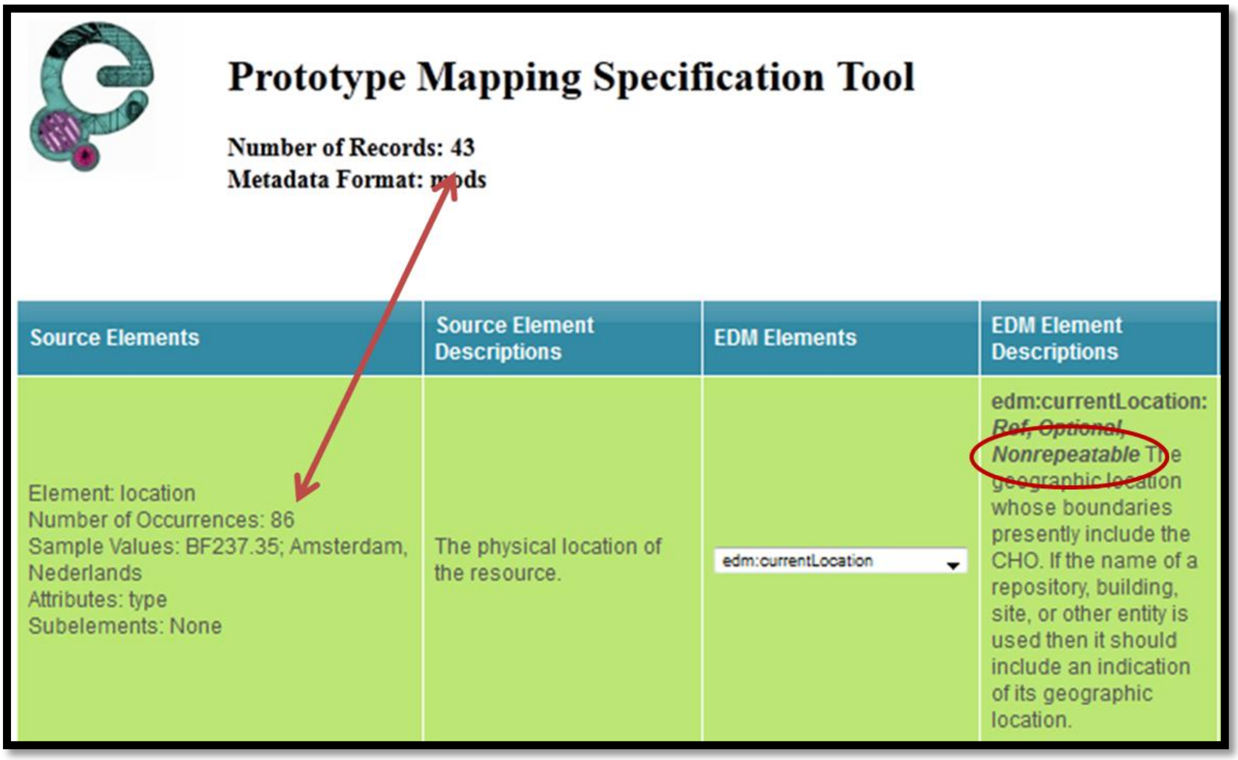

Figure 6.5: Identification of Repeatable Source Elements

Another intent issue comes into play especially in tasks that incorporate multiple source elements (many-to-one combination situations). The design of the mapping tool lends itself better to considering each source element individually (as every source element has its own individual row). However, in many-to-one combination situations, multiple rows in the table must be examined simultaneously. The representative task provides a rather simple instance – in a hypothetical source schema that separates the author into *givenName* and *lastName*, both elements should be mapped into the same *dc:creator* field. By looking at the element information and definition for *givenName* – and even just the element name itself – the user could easily deduce that this is only one part of a person's name (more specifically, the author), and, examining the EDM element options, realize that the field for main author (*dc:creator*) is not split into first and last name elements. However, if the user realizes they might need to combine two separate source elements into one target field, there can be confusion as to how to proceed, especially as the user attempts to compare the element information and definition of two or more source elements to the definitions of (potentially multiple) EDM elements. This can be even more difficult if the source element rows are not near each other within the table (which is a possibility).

Other intent issues encountered included:

- In an overlap situation, where the source element and the chosen target element have divergent definitions and values, the user might not explain this fully in the note section provided.
- The user may not ensure that all mandatory target elements are met.
- If the source element should be separated into two different target elements (one-to-many-unique situation), the user may become very confused on how to proceed – as there is only one dropdown box per source element, and only one EDM element may be selected at a time.

The mapping tool design attempts to provide a vast amount of information about every element in both schemas. However, how this information is displayed – densely and separated based on source elements – occasionally resulted in user intent issues during the cognitive walkthrough in certain mapping situations.

6.3.2 Action Visibility Issues

An action visibility issue was identified in the cognitive walkthrough when the user would be unaware of the presence of the action for a given step. Action visibility is addressed in the cognitive walkthrough by answering the question *will the user notice that the correct action is available?* This does not necessarily indicate that the user is aware that the action is the correct one for the step they are attempting to complete, merely that they are aware that the action is *there*.

There were five action visibility issues identified by the cognitive walkthrough. Three of these issues occurred when the user was expected to reference back to the metadata information for the source element that is located in the first cell of the applicable row after examining potential target elements for mapping. The cluttered nature and organization of the first cell could reduce the visibility of the information they are expected to reference in certain action steps – especially noticing details concerning the sample values.

The remaining issues concerned the selection process for potential target elements, especially if the source element lacks a syntactic (and possible semantic) similarity to any EDM element names. A first time user of the mapping tool may not realize that the EDM element definitions appear when an element is selected from the dropdown menu, though, they will likely discover this very quickly. If there are no obvious possible mappings (based on the EDM element names), the user would be forced to select each EDM element individually to check the definition, which could be extremely time-consuming.

The visibility issues encountered in the cognitive walkthrough indicated that the information and actions in the mapping tool were mostly visible; however, the dense and occasionally cluttered text contained within specific cells in the display, in theory, could cause certain significant details to be difficult to detect.

6.3.3 Action-Intent Connection Issues

An action-intent connection issue was identified in the cognitive walkthrough if the user might not connect the correct action with the intended effect within an action step, and was extrapolated in the cognitive walkthrough with the question *will the user associate the correct action with the effect they are trying to achieve?* This question assumes that the user is attempting to achieve the correct effect for the step, and that the action is adequately visible, and then examines whether the user would know to perform the action to complete the step. (When performing the cognitive walkthrough, every successive step/question is answered under the assumption that the previous steps/questions were successful. Therefore, even if the step resulted in failures in the first two questions, the action-intent connection question is answered as if the user was attempting to achieve the correct effect and could easily locate the action.)

The cognitive walkthrough encountered 16 connection issues. A recurring issue resulted from the tool's heavy reliance on the Notes field for specifying many of the mapping situations. When a user has decided on a mapping specification outside the simple One-to-One situation (e.g. that a single subject element should be expanded into multiple *dc:subject* elements, that two source elements should be combined into the same field, etc.), the Notes field often must be used to explain this mapping. In many cases, the user may anticipate the presence of a more concrete technique to stipulate the mapping specification. As such, although the Notes field for each source element row is highly visible, the user could conceive its utilization extends merely to non-codifiable information (i.e. in an overlap situation when some source values map absolutely to the target element – but others do not – and there is no way to distinguish between these values in the markup[13]); to explanations about the mapping (i.e. *why* a particular mapping was chosen); or to notations outlining collection-constant values (i.e. if the data provider is the same for all records).

The combination of multiple source elements into single EDM fields again advanced concerns in the connection portion of the cognitive walkthrough. Not only could referencing multiple rows concurrently cause intent issues – providing ambiguities on potential task procedure – but it could prevent the user from comprehending how and where within the mapping tool to notate specifications in these situations. The user could question whether the specification should be demarcated in only one or in both source element rows.

The surfeit of textual data contained within each source element row could conceivably cause action-intent connection issues, as well. The overwhelming amount of information presented – and how it is compactly presented – could prevent the user from identifying key pieces of information that will inform their mapping. For instance, while the user might be aware that sample values are provided, in the glut of text, they may not return to and dissect these values throughout the mapping process.

Finally, when the source element in question lacks any obvious EDM equivalents the user could struggle in target element selection. This issue was specifically encountered in the One-to-None and Overlap situations, however, could easily occur in any mapping situation depending upon source element names. If the source element has a syntactic similarity to one

[13] XSL, which is the stylesheet language that would eventually be used to transform the original metadata into EDM following the mapping specification, is very capable of manipulating XML documents. Using the query language XPath, XSL can make very precise 'demands' in its transformations. Element positions, lengths and segments of element values, and attribute names and values are only a few of the ways XSL can access XML data. However, XSL has limits and cannot make semantic distinctions. Therefore, if only some values of a source element fit into the definition of the target element, and there is no way to delineate between these sets in a quantifiable way, the distinction would be non-codifiable.

or more of the EDM elements, the user is met with an obvious course. These elements may not be part of the correct mapping, but their definitions will often lead in the right direction. For example, a source element *date* has a syntactic similarity to the EDM element dc:date. By selecting this target element, the user is instructed to use *dcterms:issued* or *dcterms:created* instead, if applicable. Also, the user could identify a semantic similarity between the source element name and definition and a target element name, which is indicative of a potential mapping (e.g. a source element *author* would be semantically linked to the target element *dc:creator*).

However, if the source element name has neither a syntactic or semantic relationship with any EDM elements, the user could be overwhelmed with the number of possibilities and unsure how to complete the action.

Many of the action-intent connection issues build upon problems identified by intent and visibility issues – consulting multiple source element rows; absorbing exorbitant text presented in limited space; sifting through profuse EDM elements when the source element name, metadata information, and definition provide little to no guidance. However, perhaps the most important issue identified was not a compounded problem throughout the action step categories. The mapping tool's heavy reliance on the Notes field for multiple and varied mapping situations is foremost an issue of connection – as the user's intentions were generally clear (specifying within the tool the chosen mapping for the source element) and the required action quite visible in the source element row (typing in the Notes field). That the user likely would not align their goal with the tool's avenue for completion exemplifies a disconnect between intent and action, as well as user and tool.

6.3.4 Progress Issues

The cognitive walkthrough indicated a progress issue when the user could be unsure whether they are moving towards completion of the task. This is examined by answering the question *Will the user see that progress is being made toward the solution of their task?*

There were eight progress issues identified by the cognitive walkthrough. All of the progress issues built upon problems encountered in previous categories within the action steps. These included the use of the Notes field to specify certain mappings; switching between multiple source element rows; and the lack of a target element match.

The heavy reliance of the tool on the Notes field was also a factor in progress issues. Even if the user decided to utilize the Notes field to outline certain aspects of the mapping specification, they could conceivably find this solution wanting. The user could be skeptical that the element's specification is completed in this manner.

When the user would be expected to evaluate multiple rows at once due to a Many-to-One mapping situation, action steps indicating switching between and referencing the separate rows could leave the user uncertain about their progress – moving back and forth between rows may not indicate to the user that they are nearing task completion.

Finally, the user may be tentative in an Unmatched Source Element situation to commit to stating there is no match among the target elements. They may be unwilling to confirm task completion and maintain there is no match for the source element among the EDM elements.

The progress issues shown by the cognitive walkthrough were all reiterations of similar problems encountered in previous issue categories. However, this does not lessen the importance of these issues.

6.4 Summary

There were multiple usability issues encountered in the cognitive walkthrough analysis. One of the most prominent concerns was the user's ability to identify and use the information as provided. In addition, the significant use of the Notes field for explaining mapping situations could cause confusion for the user. Navigating between multiple source element rows when required was also found to be an issue.

Chapter 7: Conclusion and Discussion

7.1 Conclusion

What are the functional requirements for creating a browser-based prototype for mapping cultural heritage schema to the Europeana metadata standard?

The functional requirements for the creation of the mapping prototype were outlined in depth in Chapter 5. The requirements of the mapping tool were based on those element components outlined by St. Pierre & LaPlant (1998) and Chan & Zeng (2006):

- A semantic definition;
- Occurrence constraints;
- Structural constraints;
- Value constraints; and
- Actual source metadata values.

This required an input method for the metadata to be mapped (in order to allow for the examination of real values) and a way to identify all source elements so their definitions may be displayed.

The focal point of the prototype was the table display mapping specification tool, which was contained in an HTML page. In order to analyze the usability of this display the prototype was housed on a server, to which XML metadata files could be uploaded to display in the mapping tool. To achieve the upload and subsequent display of the metadata in the tool, PHP script and XSLT were required. In order to display the source element definitions for all elements present in the metadata, a separate XML file had to be created from which the XSL could extract the relevant information. For the prototype, this file only included a select number of schemas for which element definitions where available to present in the mapping tool display.[14]

The mapping tool screen required a way to map the source elements to the chosen EDM elements. As the source elements were arranged into rows in the mapping, a dropdown menu containing all EDM elements – categorized by their core classes – was included in each row.

[14] For a completed mapping tool, element definitions for as many schemas as possible would be included. Whether or not the exact same method for storing and retrieving these definitions would be used is uncertain, and was not a focal point of the research.

In order to display the EDM definitions, Javascript was used to make this text dynamically appear when a given EDM element was chosen. A text box was included with each row, to allow for addition notes on the specification. These conditions allowed for the prototype mapping tool to function as required in the browser, in order to complete the usability analysis through the cognitive walkthrough.

To what extent is the prototype able to complete potential element mapping situations from the source metadata formats into the first implementation of the Europeana Data Model (EDM) format?

The prototype was technically capable of allowing for the specification of each of the eleven potential mapping situation tasks recorded in the cognitive walkthrough. However, for many of the mapping situations, the tool relied upon the Notes field to fully specify the mapping:

1. One-to-One: The one-to-one, equivalent mapping was very simple to specify in the mapping tool.
2. One-to-Many: Identical Target Elements: If a single source element value should be separated into multiple instances of the same target element, how this split is meant to be made must be specified in the Notes section.
3. One-to-Many: Unique Target Elements: If a single source element value should be separated into multiple different EDM elements, not only must the split itself be specified in the Notes field, but the tool allows only one actual EDM element to be selected at a time, which results in any additional elements being typed out in the Notes field with explanations of how the value should be split between them.
4. Many-to-One: Combination: This situation requires the combination of multiple source element values into a single EDM element value. The notes field must be used to specify how the concatenation of the values will be performed.
5. Many-to-One: Choice: If a source element being mapped to a non-repeatable EDM element occurs multiple times in a single record and the user decides to choose only one to map, instead of concatenating the values, the Notes field must be used to specify how this choice is to be made.
6. One-to-None: This situation can be completed in the mapping tool by not selecting an EDM element to map to.
7. Mandatory Target Elements Unmatched: The mapping tool provides extra rows at the bottom for any mandatory EDM elements not matched to any source elements. These

rows require the use of the Notes field, then, to specify from where the required element values will come.

8. Overlap: If the source element values only fit appropriately into the chosen EDM element definition a portion of the time, the Notes field could be used to provide more details on this divergence.
9. Conversion Combinations: Conversion combination situations are extremely varied in scope, and as such, it would be impossible to ensure that all potential combinations could be
10. Content Requirement – Data Type/Value Range: If the target element has a data type or value range requirement, and the source element values do not meet this requirement, the Notes field must be used to explain how the source values will be normalized.
11. Content Requirement – Controlled Vocabulary: For the two EDM elements that have a controlled vocabulary, the Notes field must be used to indicate how the source values will be translated into the correct controlled vocabulary values.

The final three situations – Single vs. Multiple Objects, Logical Views, and Hierarchy – were not analyzed, as they are broader situations concerned with records as a whole.

Therefore, while the prototype was able to handle all eleven of the element mapping situations, it was largely through the use of the open text box. For some instances of the mapping situations, this may be the most appropriate way to clarify a specification. For other situations, perhaps it would be possible to modify the design to minimize the reliance on the Notes field, as this reliance resulted in usability issues during the cognitive walkthrough.

What are the usability issues associated with this HTML table-based structure for a mapping tool to create specifications from local cultural heritage collections into the first specification of EDM?

The research uncovered numerous usability issues with the proposed mapping tool design. All mapping situations examined except for the one-to-one situation encountered some usability issues with the mapping tool during the cognitive walkthrough. While the mapping tool was successful in many steps of the eleven situation tasks, the usability issues encountered in the cognitive walkthrough of the prototype must be addressed in further development of the mapping tool.

Element information display At times, the tool's display provided both too much and too little information. The mass of information, which was considered to be the most positive aspect of the proposed design, also served as a usability deterrent. As a significant amount of source element information was contained within the first two columns of the prototype design, this information could be very difficult to identify and address throughout the mapping processes.

While the tool does include sample values for each of the source elements, they are buried within the information provided in the mapping tool. In addition, the small number of sample values that are displayed may not provide as much benefit as originally anticipated. This cluster of information may make it difficult for the user to identify key components of the mapping to inform their decision.

There are a few EDM elements that are non-repeatable. When mapping a source element to a non-repeatable target element, the mapper must know whether the source element repeats. Otherwise, this situation could lead to data loss and/or result in an invalid specification. However, the tool does not provide an absolute way to detect the number of element occurrences within an individual record.

Notes field As discussed in the previous research question response, the Notes field is heavily relied upon in the mapping tool prototype. In some of the situations' steps, the use of the Notes field was an obvious action. For instance, the user would likely expect to use a text area to explain the relationship between the source and target element definitions in an overlap situation. In other situations, however, the use of the Notes field to notate rather quantifiable specifications, would not be expected by the user.

Multiple source row consideration Finally, the mapping tool's display of information by source element rows does not lend itself well to many-to-one situations invovling multiple source elements. This is an especially important issue, as the mapping tool design is really built upon this way of displaying the source metadata information, and is not well designed to consider multiple source elements simultaneously.

The prototype mapping tool as created was able to assist in the completion of the cognitive walkthrough for usability analysis. While the prototype was technically able to complete the eleven mapping situations analyzed in the cognitive walkthrough, in many of the mapping situations required the use of the Notes field to fully specify the element mapping.

For a mapping tool design to incorporate the five elements components necessary to consider in mapping situations, there must be a careful balance between too much and not enough information and the display of this information must be very carefully considered. The mapping tool prototype's presentation of source element information was cluttered, resulting in usability issues, and still, at times the information provided could be lacking. The prototype tools reliance on the Notes field to specify many mapping situations – even those that might be better specified in more quantifiable input methods – presented usability issues in several tasks. Finally, the issue that is closely tied to the very basic design of the mapping tool is when multiple source elements must be mapped to a single EDM element. The crux of the mapping tool design is the table structure with a row for each source element, and this is design as implemented in the prototype is ill-suited for examining two source elements simultaneously.

7.2 Discussion

For this mapping tool design to be practicable, there would have to be several changes to address those usability issues identified during the cognitive walkthrough. The exact form these changes might take was not directly addressed by the research questions, and therefore not a focal point of the research. Instead, the cognitive walkthrough merely exposed issues which should be addressed.

Element values are an interesting area that must be carefully considered – and part of the greater issue of element information display – not only in this mapping tool to EDM, but in any mapping tool that seeks to map an instantiation of a metadata schema. The element values should be a part of a quality mapping, however, there is difficulty in presenting them as a part of the tool. If only a small sample of the values are presented, as with the prototype mapping tool design, they may not provide adequate guidance in the mapping process. However, if more or all values were presented, the tool would be inundated with even more textual information, perhaps overwhelming the user. Perhaps an avenue for exploration would be to have all of the element values conditionally visible (i.e. via hypertext which opens an additional window with all values) or in a scrollable text area.

One of the issues encountered in the cognitive walkthrough was the inability to decisively determine whether or not an element was present multiple times in any of the source records. While both the total number of element occurrences and the total number of records were presented, the comparison of these to values does not give any absolute assurance whether or not an element has been repeated in a record. The tool ought to more clearly indicate whether the element has been repeated in any of the source records.

Further research of this and other real metadata-based mapping tools could extend to element attributes, which were not addressed fully in the situations. Metadata schema can rely upon element attribute values to differentiate value connotations. While the mapping tool prototype displayed all element attributes present in the metadata, it did not display all possible values for each element. The usability of the tool in more complex record level situations – Single vs. Multiple Objects, Logical Views, and Hierarchy – should also be examined.

A very important area of investigation is that of concrete delineation of the element mapping specifications. Though all eleven of the situations analyzed in the cognitive walkthrough could technically be expressed in the mapping tool, many of the more complex situations relied heavily on the use of the Notes field to provide exact specification instructions. Though some use of this Notes field may be unavoidable, more definitive ways of expressing these situations ought to be explored, as well, for two main reasons. First, the cognitive walkthrough showed that use of the Notes field to explain certain situations could be confusing to the user. Second, an extension to this mapping tool in the future could be an automatic creation of either the XSLT document that would be used to transform the data into EDM; the uploaded metadata transformed into EDM XML; or both. The creation of these documents would require the mapping specification to be extremely exact and quantitative in nature; there could be no ambiguity, as the computer must be able to identify the user's intent.

The prototype itself did not verify that required EDM elements had been assign values somewhere within the specification, which could be beneficial to ensure the user specifies these elements. It could also be helpful to delineate the required elements in the list of all of the EDM elements, further prompting the user to address these elements. The tool may also dynamically recognize those required elements that have been assigned, and only display rows below the source element table for those required elements that remain unassigned.

The necessity to be able to consider multiple source elements together in certain situations could result in research of how to allow this in a table display. Avenues that may be considered could include being able to rearrange table rows and/or being able to easily link and unlink rows.

While this research places all usability issues solely on the mapping tool design, a tangential area for research is how EDM could potentially be improved in relation to the mapping situations in order to avoid complex situations inasmuch as possible. However, this possibility may not be particularly fruitful, as EDM has largely been focused on flexibility while still striving for homogeneity in elements. As shown in the explanation of the mapping situations in an EDM context, the likelihood of many situations occurring is greatly minimized by the construct of EDM. There are very few constraints on element content values and number of allowable occurrences; and EDM is largely based on Dublin Core, which is meant to be a generalizable schema.

References and Works Cited

Brooks, F. (1995). *The mythical man-month.* Boston: Addison-Wesley.

Caplan, P. (2006, July). Preservation metadata. (S. Ross, & M. Day, Eds.) *DCC Digital Curation Manual*, 1-26.

Chan, L., & Zeng, M. (2006, June). Metadata interoperability and standardization – A study of methodology part I. *D-Lib Magazine, 12*(6).

Costanza, J., Knight, r., & Liu-Spencer, H. (2009). Metadata implementation for building cross-institutional repositories: Lessons learned from the Liberal Arts Scholarly Repository (LASR). *Journal of Library Metadata*, 152-166.

Dekkers, M., Gradmann, S., & Molendijk, J. (2011, October 5). *Final technical & logical architecture and future work recommendations.* Retrieved February 28, 2012 , from europeana.eu: http://pro.europeana.eu/documents/10602/370691/D3.4+final.pdf

Doerr, M., Gradmann, S., Hennicke, S., Isaac, A., Meghini, C., & van de Sompel, H. (2010). The Europeana data model (EDM). *World Library and Inforamtion Congress: 76th IFLA General Conference and Assembly.* Gothenburg, Sweden.

Downey, A., Elkner, J., & Meyers, C. (2008). *How to think like a computer scientist: Learning with Python.* Wellesley, Massachusetts: Green Tea Press.

Europeana. (2011). *The Europeana data model for cultural heritage.* Hentet February 27, 2012 fra europeana.eu: http://pro.europeana.eu/c/document_library/get_file?uuid=ef2baffc-f078-41d9-be5f-76a3427f198f&groupId=51031

Europeana. (2011b, October 26). *Europeana data model primer.* Retrieved February 28, 2012, from Europeana: http://version1.europeana.eu/c/document_library/get_file?uuid=48b552e1-c71d-4f3f-a1ae-c1929cb7de76&groupId=10605

Europeana. (2011c, October 27). *Europeana Data Model mapping guidelines.* Hentet January 15, 2012 fra europeana.eu: http://pro.europeana.eu/documents/51031/605549/EDM_guidelines_v1.0.pdf

Europeana. (2011d, October 27). *Indicative targets for minimum content contribution to Europeana per Member State*. Retrieved March 13, 2012, from Europeana Professional: http://pro.europeana.eu/documents/10602/15280/EC+-+Indicative+targets+per+Member+State.pdf

Europeana. (2011e). *Facts & figures*. Hentet April 2, 2012 fra Europeana Professional: http://pro.europeana.eu/web/guest/about/facts-figures

Gerard, S., & Singh, M. (2010). Protocol refinement: Formalization and verification. In A. K. Chopra, A. Artikis, J. Bentahar, & F. Dignum (Ed.), *Proceedings of the AAMAS Workshop on Agent Communication.*

Harold, E., & Means, W. (2004). *XML in a nutshell* (3rd ed.). Sebastopol, CA: O'Reilly Media.

Haslhofer, B., & Klas, W. (2010, February). A survey of techniques for achieving metadata interoperability. *ACM Computing Surveys, 42*(2), 7:1-7:37.

Intner, S., Lazinger, S., Weihs, & J. (2006). *Metadata and its impact on libraries.* Westport, Connecticut: Libraries Unlimited.

John, B., & Packer, H. (1995). *Learning and using the cognitive walkthrough method: A case study approach.* Hentet April 15, 2012 fra CHIU95Proceedings: http://www.sigchi.org/chi95/proceedings/papers/bej1bdy.htm

Kolaitis, P. (2005). Schema mappings, data exchange, and metadata management. *PODS 2005 June 13-15*, 61-75.

Lagoze, C., & Van de Sompel, H. (2001, June 24-28). The Open Archives Initiative: Building a low-barrier interoperability framework. *JCDL'01*, 54-62.

Lagoze, C., & Van de Sompel, H. (2008, October 17). *ORE specification - Abstract data model*. Hentet February 28, 2012 fra Open Archives Initiatve: http://www.openarchives.org/ore/1.0/datamodel

Mahatody, T., Sagar, M., & Kolski, C. (2010). State of the art on the cognitive walkthrough method, its variants and evolutions. *International Journal of Human-Computer Interaction, 26*(8), 741-785.

Miller, S. (2011). *Metadata for digital collections: A how-to-do-it manual.* London: Facet.

Mowat, J. (2002, April 19). *Cognitive walkthroughs: Where they came from, what they have become, and their application to EPSS design.* Hentet April 15, 2012 fra www.herridgegroup.com/pdfs/cognitive.pdf

Nielsen, J. (1995). Usability inspection methods. *CHI '95 Mosaic of Creativity* (ss. 377-378). Denver: ACM.

NISO. (2004). *Understanding metadata.* Retrieved 01 29, 2012, from http://www.niso.org/publications/press/UnderstandingMetadata.pdf

NISO. (2007, December). *A framework of guidance for building good digital collections.* Retrieved February 5, 2012, from niso.org: http://www.niso.org/publications/rp/framework3.pdf

Noy, N., & Musen, M. (2002). Evaluating ontology-mapping tools: Requirements and experience. *13th International Conference on Knowledge Engineering and Knowledge Management*, 1-14.

Paepcke, A., Chang, C., Garcia-Molina, H., & Winograd, T. (1998, April). Interoperability for digital libraries worldwide. *Communications of the ACM, 41*(4), 33-43.

Pickard, A. (2007). *Research methods in information.* London: Facet.

Pinelle, D., & Gutwin, C. (2002). Groupware walkthrough: Adding context to groupware usability. *Proceedsing of ACM CHI*, (ss. 455-462).

Pressman, R. (2010). *Software engineering: A practitioner's approach* (7th International ed.). Boston: McGraw-Hill Higher Education.

Sears, A., & Hess, D. (1999). Cognitive walkthroughs: Understanding the effect of tas-description detail on evaluator performance. *International Journal of Human-Computer Interaction*, *11*(3), 185-200.

Sommerville, I. (2011). *Software engineering* (9th ed.). Harlow: Addison-Wesley.

Spaccapietra, S., Parent, C., & Dupont, Y. (1992). Model independent assertions for integration of heterogeneous schemas. *VLDB Journal, 1*(1), 81-126.

St. Pierre, M., & LaPlant, W. (1998, October 15). *Issues in crosswalking content metadata standards.* Hentet February 5, 2012 fra niso.org: http://www.niso.org/publications/white_papers/crosswalk/

TEI Consortium. (2012). *TEI: Text Encoding Initiative*. Hentet February 19, 2012 fra www.tei-c.org: http://www.tei-c.org/index.xml

Tolk, A., & Muguira, J. (2003, September). The levels of conceptual interoperability model. *2003 Fall Simulation Interoperability Workshop*, 1-12.

U.S. Department of Defense. (1998). Levels of Information Systems Interoperability (LISI). *Architectures Working Group*.

Vullo, G., Innocenti, P., & Ross, S. (2010). Towards a digital library policy and quality interoperability framework: The DL.org project. *New Review of Information networking, 15*, 29-53.

W3C. (2004, February 10). *Resource description framework*. Hentet February 19, 2012 fra W3C Semantic Web: http://www.w3.org/RDF/

Wharton, C., Rieman, J., Lewis, C., & Polson, P. (1993). *The cognitive walkthrough method: A practioner's guide.* Retrieved April 30, 2012, from University of Colorado at Boulder Institute of Cognitive Science: http://ics.colorado.edu/techpubs/pdf/93-07.pdf

Wharton, C., Rieman, J., Lewis, C., & Polson, P. (1994). The cognitive walkthrough method: A practitioner's guide. I J. Nielsen, & R. Mack (Red.), *Usability instpection methods* (ss. 105-140). New York: Wiley & Sons.

Woodely, M. (2008). Crosswalks, metadata harvesting, federated searching, metadsearching: Using metadata to connect users and information. I M. Baca (Red.), *Introduction to metadata:* (2nd. utg., ss. 38-62). Los Angeles, CA: Getty Publications.

Yuan, J., Bahrami, A., Wang, C., Murray, M., & Hunt, A. (2006). A semantic information integration tool suite. *Very Large Data Base 2006, September 12-15, Seoul, Korea*, 1171-1174.

Zeng, M., & Chan, L. (2006, June). Metadata interoperability and standardization -- A study of methodology part II. *D-Lib Magazine, 12*(6).

Zeng, M., & Qin, J. (2008). *Metadata.* London: Facet.

Zeng, M., & Xiao, L. (2001). Mapping metadata elements of different format. *E-Libraries 2001, Proceedings, May 15-17, 2001, New York* (ss. 91-99). Medford, NJ: Information Today, Inc.

Zimmermann, H. (1980). OSI reference model--The ISO model of architecture for open systems interconnection. *IEEE Transactions on Communication, COM-28*(4), 425-432.

Appendix A: Elements in Core Classes in the First Implementation of EDM

A recreation of the EDM core class tables in the EDM Mapping Guidelines V1.0 (Europeana, 2011c).

EDM Element	Content	Required	Repeatable	Description
edm:ProvidedCHO				
owl:sameAs	Ref	No	Yes	Use to point to your own (linked data) representation of the object, if you have already minted a URI identifier for it. It is also possible to provide URIs minted by third-parties for the object.
dc:contributor	Literal/Ref	No	Yes	For contributors to the CHO. If possible supply the identifier of the contributor from an authority source.
dc:overage	Literal/Ref	No	Yes	The spatial or temporal topic of the CHO. Use the more precise dcterms:spatial or dcterms:temporal properties if the data will support it. One of dc:coverage or dc:subject or dc:type or dcterms:spatial must be provided.
dc:creator	Literal/Ref	No	Yes	For the creator of the CHO. If possible supply the identifier of the creator from an authority source.
dc:date	Literal/Ref	No	Yes	Use for a significant date in the life of the CHO. Consider the subproperties of dcterms:created or dcterms:issued.
dc:description	Literal/Ref	No	Yes	A description of the CHO. Either dc:description or dc:title must be provided.
dc:format	Literal/Ref	No	Yes	Use for the format of a born digital cultural heritage objects or for other terms applied to the CHO.
dc:identifier	Literal	No	Yes	An identifier of the original CHO.
dc:language	Literal	No	Yes	The language of text CHOs and also for other types of CHO if there is a language aspect. Mandatory for TEXT objects, strongly recommended for other object types with a language element.
dc:publisher	Literal/Ref	No	Yes	The name of the publisher of the CHO. If possible supply the identifier of the publisher from an authority source.
dc:relation	Literal/Ref	No	Yes	The name or identifier of a related resource, generally used for other related CHOs. Cf edm:isRelatedTo
dc:rights	Literal/Ref	No	Yes	Use to give the name of the rights holder of the CHO if possible or for more general rights information. Note the difference between this property and the use of the controlled edm:rights property which relates to the digital objects (see WebResource and Aggregation tables).
dc:source	Literal/Ref	No	Yes	The source of the original CHO. This property should no longer be used for the name of the content holderfor this, see edm:dataProvider in the ore:Aggregation table below.
dc:subject	Literal/Ref	No	Yes	The subject of the CHO. One of dc:subject or dc:coverage or dc:type or dcterms:spatial must be provided.
dc:title	Literal	No	Yes	The title of the CHO. Either dc:title or dc:description must be provided.
dc:type	Literal/Ref	No	Yes	The nature or genre of the CHO. Ideally the term(s) will be taken from a controlled vocabulary. One of dc:type or dc:subject or dc:coverage or dcterms:spatial must be provided.
dcterms:alternative	Literal	No	Yes	Any alternative title of the CHO including abbreviations or translations.
dcterms:conformsTo	Literal/Ref	No	Yes	An established standard to which the CHO conforms.
dcterms:created	Literal/Ref	No	Yes	The date of creation of the CHO.
dcterms:extent	Literal/Ref	No	Yes	The size or duration of the CHO.
dcterms:hasFormat	Literal/Ref	No	Yes	A resource related to the CHO that is substantially the same as the CHO but in another format.
dcterms:hasPart	Literal/Ref	No	Yes	A resource that is included either physically or logically in the CHO.
dcterms:hasVersion	Literal/Ref	No	Yes	Another resource that is a version, edition or adaptation of the CHO demonstrating substantive changes in content rather than format.
dcterms:isFormatOf	Literal/Ref	No	Yes	Another resource that is substantially the same as the CHO but in another format.

dcterms:isPartOf	Literal/Ref	No	Yes	A resource in which the CHO is physically or logically included. This property can be used for objects that are part of a hierarchy and will be used to support an appropriate display in the portal. For that purpose it will be necessary to supply an identifier as the value.
dcterms:isReferenceBy	Literal/Ref	No	Yes	Another resource that references, cites or otherwise points to the CHO.
dcterms:isReplacedBy	Literal/Ref	No	Yes	Another resource that supplants , displaces, or supersedes the CHO.
dcterms:isRequiredBy	Literal/Ref	No	Yes	Another related resource that requires the CHO to support its function, delivery or coherence.
dcterms:issued	Literal/Ref	No	Yes	Date of formal issuance or publication of the CHO.
dcterms:isVersionOf	Literal/Ref	No	Yes	Another resource of which the CHO is a version, edition or adaptation, demonstrating substantive changes in content rather than format.
dcterms:medium	Literal/Ref	No	Yes	The material or physical carrier of the CHO.
dcterms:provenance	Literal/Ref	No	Yes	statement of changes in ownership and custody of the CHO since its creation. Significant for authenticity, integrity and interpretation.
dcterms:references	Literal/Ref	No	Yes	Other resources referenced, cited or otherwise pointed to by the CHO.
dcterms:replaces	Literal/Ref	No	Yes	A related resource that is supplanted, displaced, or superseded by the CHO.
dcterms:requires	Literal/Ref	No	Yes	Another resource that is required by the described resource to support its function, delivery or coherence.
dcterms:spatial	Literal/Ref	No	Yes	Spatial characteristics of the CHO. i.e. what the CHO represents or depicts in terms of space (e.g. a location, coordinate or place). Either dcterms:spatial or dc:type or dc:subject or dc:coverage must be provided.
dcterms:tableOfontents	Literal	No	Yes	A list of subunits of the CHO.
dcterms:temporal	Literal/Ref	No	Yes	Temporal characteristics of the CHO. i.e. what the CHO is about or depicts in terms of time (e.g. a period, date or date range.)
edm:type	Literal	Yes	No	The provided object is one of the types accepted by Europeana and will govern which facet it appears under in the portal - TEXT, VIDEO, SOUND, IMAGE, 3D
edm:currentLocation	Ref	No	No	The geographic location whose boundaries presently include the CHO. If the name of a repository, building, site, or other entity is used then it should include an indication of its geographic location.
edm:isNextInSequence	Ref	No	No	The identifier of the preceding object where both objects are part of the same overall resource. Use this for objects that are part of a hierarchy or sequence to ensure correct display in the portal.
edm:WebResource				
dc:rights	Literal/Ref	No	Yes	Use for the name of the rights holder of this digital representation if possible or for more general rights information. Note the difference between this property and the use of the mandatory, controlled edm:rights property below.
edm:rights	Literal/Ref	No	No	The value in this element will indicate the usage and access rights that apply to this digital representation. For the first implementation of EDM only the rights associated with the ore:Aggregation class will be implemented. Later implementations will be able to use the edm:rights associated with individual WebResources so it is strongly recommended that a value is supplied for this property for each instance of a WebResource. See also edm:rights in the ore:Aggregation table.

ore:Aggregation				
edm:aggregatedCHO	Ref	Yes	No	The identifier of the source object e.g. the Mona Lisa itself. This could be a full linked open data URI or an internal identifier.
edm:dataProvider	Literal/Ref	Yes	No	The name or identifier of the data provider of the object (i.e. the organisation providing data to an aggregator). Identifiers will not be available until Europeana has implemented its Organisation profile.
edm:hasView	Ref	No	Yes	The URL of a web resource which is a digital representation of the CHO. This may be the source object itself in the case of a born digital cultural heritage object. edm:hasView should only be used where there are several views of the CHO and one (or both) of the mandatory edm:isShownAt or edm:isShownBy properties have already been used. It is for cases where one CHO has several views of the same object. (e.g. a shoe and a detail of the label of the shoe).
edm:isShownBy	Ref	No	No	The URL of a web view of the object. Either edm:isShownAt or edm:isShownBy is mandatory. For the rights that will apply to previews please see edm:rights below.
edm:isShownAt	Ref	No	No	The URL of a web view of the object in full information context. Either edm:isShownAt or edm:isShownBy is mandatory. For the rights that will apply to previews please see edm:rights below.
edm:object	Ref	No	No	The URL of a representation of the CHO which will be used for generating previews for use in the Europeana portal. This may be the same URL as edm:isShownBy. See Europeana Portal Image Guidelines (http://version1.europeana.eu/web/guest/technical-requirements) for information regarding the specifications of previews.
edm:provider	Literal/Ref	Yes	No	The name or identifier of the provider of the object (i.e. the organisation providing data directly to Europeana). Identifiers will not be available until Europeana has implemented its Organisation profile.
dc:rights	Literal	No	Yes	Ideally this should be applied to the edm:WebResource or the edm:ProvidedCHO. It is included here for the conversion of data from ESE where it is not known which object the rights apply to.
edm:rights	Ref	Yes	No	This is a mandatory property and the value given here should be the rights statement that applies to the digital representation at the URL given in edm:object or edm:isShownAt/By. The value should be taken from one of those listed in the Europeana Rights Guidelines (http://version1.europeana.eu/web/guest/technical-requirements) The rights statement given in this property will also apply to the previews used in the portal and will be the source of(1) the entry in the Rights facet in the portal, (2) the license badge that appears under the preview on the result page. Where there are several web resources attached to one edm:ProvidedCHO the rights statement given here will be regarded as the "reference" value for all the web resources so a suitable value should be chosen if the rights statements vary between different resources. In future implementations it is hoped to handle rights statements for separate web resources associated with one CHO separately.

Appendix B: XSLT to Transform Source Metadata for Mapping Tool

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 exclude-result-prefixes="xsl oai xsi exsl"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:exsl="http://exslt.org/common"
    xmlns:oai="http://www.openarchives.org/OAI/2.0/"
    extension-element-prefixes="exsl"
    xsi:schemaLocation="http://www.europeana.eu/schemas/ese/
http://www.europeana.eu/schemas/ese/ESE-V3.3.xsd">

<!-- These removes the inconsistent, extraneous space between any elements, and
format the result document like an xml document (with indents) -->
<xsl:strip-space elements="*"/>
<xsl:output method="html" indent="yes"/>

<!-- Variables used to reference and include element definitions for the
elements in the metadata. -->
    <xsl:variable name="metadata" select="oai:OAI-
PMH/oai:request/@metadataPrefix"></xsl:variable>
    <xsl:variable name="desc_doc" select="document('sourceDescriptions.xml')"/>
    <xsl:variable name="desc" select="$desc_doc/schemas/descendant::*"/>

<!-- Populates the first column of the table with the metadata information -
name, number of occurrences, etc. -->
<xsl:template match="oai:ListRecords">
        <xsl:variable name="element"
select="oai:record/oai:metadata/*/child::node()"></xsl:variable>
       <xsl:variable name="content" select="$element[text()]"></xsl:variable>
        <xsl:variable name="wrapper" select="$element[not(text())]"/>
        <xsl:for-each select="$element">


            <xsl:if test="not(preceding::*[name()=name(current())])">
               <xsl:call-template name="elements"/>
            </xsl:if>
            <xsl:for-each select="child::*">
                <xsl:if test="not(preceding::*[name()=name(current())])">
                <xsl:call-template name="elements"/>
                </xsl:if>
            </xsl:for-each>
        </xsl:for-each>

    </xsl:template>

<xsl:template name="elements">
    <xsl:variable name="Following"
select="following::*[name()=name(current())]"/>
    <xsl:variable name="Following_att" select="$Following/@*"/>
    <xsl:variable name="all_elements" select="node()[name()=name(current())]"/>
    <xsl:variable name="first_atts" select="@*"/>


    <tr><td width="200">Element: <xsl:value-of select="name()"/><br/>
```

```
        Number of Occurrences: <xsl:value-of
select="count($Following)+1"/><br/>
        Sample Values: <xsl:if test="text()"><xsl:value-of
select="."/>;<xsl:value-of select="$Following[2]"/></xsl:if> <br/>
        Attributes: <xsl:for-each select="$first_atts">
            <xsl:variable name="att_name" select="name()"/>
            <xsl:value-of select="$att_name"/>,
        </xsl:for-each>
        <xsl:for-each select="$Following">
            <xsl:variable name="Preceding"
select="preceding::*[name()=name(current())]"/>
            <xsl:variable name="Preceding_att" select="name($Preceding/@*)"/>
            <xsl:variable name="attribute" select="@*"/>
            <xsl:for-each select="$attribute">
                <xsl:variable name="att_name" select="name()"/>
               <xsl:if test="not($Preceding/@*[name()=$att_name])">
                <xsl:value-of select="$att_name"/>,
               </xsl:if>
                </xsl:for-each>
        </xsl:for-each><br/>
        <xsl:if test="not(parent::*[parent::oai:metadata])">Parent Element:
<xsl:value-of select="name(parent::*)"/><br/></xsl:if>
        Subelements: <xsl:choose>
            <xsl:when test="child::*">
                <xsl:for-each select="child::*">
                    <xsl:value-of select="name()"/>,
                </xsl:for-each>
            </xsl:when>
            <xsl:when test="$Following/child::*">
                <xsl:for-each select="child::*">
                    <xsl:value-of select="name()"/>,
                </xsl:for-each>
            </xsl:when>
            <xsl:otherwise>None</xsl:otherwise>
        </xsl:choose>
        <br/></td>
<!-- Extracts element definitions from external XML document  -->
        <td width="200"><xsl:value-of select="$desc/descendant-or-
self::*[name()=$metadata]/descendant::*[name()=name(current())]"/></td>
<!-- Creates a dropdown menu with EDM elements for each row. The options are
imported from another document edm-drop-down2.html -->
        <td><select id="{name()}"
onChange="changeText('{name()}','display{name()}')"><xsl:copy-of
select="document('edm-drop-down2.html')"/></select></td>
        <td width="200"><div id="display{name()}"> </div></td>
<!-- Creates notes text box -->
        <td width="200"><textarea name="notes" COLS="30" ROWS="6"/></td>
    </tr>
</xsl:template>

<!-- Builds HTML page around metadata table -->
 <xsl:template match="oai:OAI-PMH">
        <html>
            <head><style type="text/css">
body {background-image:url('images/europeana-small.jpg');
background-repeat:no-repeat;}
</style><link href="tablecloth/tablecloth.css" rel="stylesheet" type="text/css"
media="screen"/>
```

```
                <script type="text/javascript"
src="tablecloth/tablecloth.js"></script></head>
            <body><div style="position: absolute; top:5px; left:125px;
width:600px; height:150px">
                <h1>Prototype Mapping Specification Tool</h1><h3>Number of
Records: <xsl:value-of select="count(//oai:record)"/><br/>
                   Metadata Format: <xsl:value-of
select="oai:request/@metadataPrefix"/></h3></div><br/><br/>
      <form>
          <table style="position: absolute; top:150px; left:5px;">
           <tr>
               <th width="200">Source Elements</th>
               <th width="200">Source Element Descriptions</th>
               <th width="200">EDM Elements</th>
               <th width="200">EDM Element Descriptions</th>
               <th width="200">Notes</th></tr>
            <xsl:apply-templates/>
              <tr><td  width="200"> </td><td  width="200"> </td><td
width="200"> </td><td  width="200"> </td><td  width="200"> <input type="submit"
value="Create Mapping Specification"/></td></tr>
          </table><br/><br/> </form>
<!-- Adds in Javascript that changes the EDM definition displayed based on the
selected element in the dropdown menu. -->
                <xsl:copy-of select="document('SelectJS2.html')"/></body>
        </html>

 </xsl:template>
```

Appendix C: Excerpt from XML Document of Source Element Definitions

```
<?xml version="1.0" encoding="UTF-8"?>
<schemas >
<mods>
    <titleInfo>Wrapper Element subelements: title, subTitle, partNumber,
partName, nonSort. Definition: A word, phrase, character, or group of
characters, normally appearing in a resource, that names it or the work
contained in it. Application: "titleInfo" is a wrapper element that contains
all subelements related to title information. Data content is contained in
subelements. &lt;titleInfo&gt; is repeated for each type attribute value. If
multiple titles are recorded, repeat &lt;titleInfo&gt;&lt;title&gt; for each.
As subelement to subject -- Definition: "titleInfo" includes a title used as a
subject. All subelements and attributes used under the top-level element
titleInfo may be used with this subelement. </titleInfo>
    <title>Subelement to titleInfo. Definition: A word, phrase, character, or
group of characters that constitutes the chief title of a resource (i.e. the
title normally used when citing the resource). Application: "title" without the
&lt;titleInfo&gt; type attribute is roughly equivalent to MARC 21 field 245,
except that it includes only the data that make up the title and does not
include the entire title area dictated by AACR2. The statement of
responsibility in MARC 21 field 245 is included in the &lt;note&gt; element
with type="statement of responsibility," not in &lt;title&gt;. Elements
considered part of the title that are separately subfielded in MARC 21 are
included here unless they have a separate subelement. For instance, the General
Material Designator (MARC 21 field 245 subfield $h) and form (MARC 21 field 245
subfield $k) may be included as part of the title, but a part name or number is
included in the subelements below.</title>
    <subTitle>Subelement to titleInfo. Definition: A word, phrase, character,
or group of characters that contains the remainder of the title information
after the title proper. Application: "subTitle" includes a subtitle when it is
desired to include it as a separate element. Alternatively, it may be included
as part of a title. It is equivalent to MARC 21 fields 242, 245, 246 subfield
$b.</subTitle>
    <partNumber>Subelement to titleInfo. Definition: "partNumber" is used for a
part or section number of a title. This is equivalent to MARC 21 fields 130,
240, 242, 243, 245, 246, 247, 730, 740 subfield $n.</partNumber>
    <partName>Subelement to titleInfo. Definition: "partName" is used for a
part or section name of a title. This is equivalent to MARC 21 fields 130, 240,
242, 243, 245, 246, 247, 730, 740 subfield $p.</partName>
    <nonSort>Subelement to titleInfo. Definition: "nonSort" begin and end tags
surround the nonfiling text which should not be regarded in sorting. It is
equivalent to the new technique in MARC 21 that uses control characters to
surround data disregarded for sorting. It is used for the same purpose as the
nonfiling indicator value that indicates the number of characters disregarded
for sorting. Punctuation may or may not be included within the non-sort value
depending upon whether it is part of the sorting or non-sorting data.</nonSort>
    <name>Wrapper Element with subelements: namePart, displayForm, affiliation,
role, description. efinition: The name of a person, organization, or event
(conference, meeting, etc.) associated in some way with the resource.
Application: "name" is a wrapper element that contains all subelements related
to name information. It is equivalent to the MARC 21 1XX and 7XX fields or
Creator and Contributor in Dublin Core. Role values may be used to indicate the
particular relationship between the name and the resource. The type of name
(personal, corporate, conference) may be indicated, although this is not
required. As subelement to subject -- Definition: "name" includes a name used
as a subject. All subelements and attributes used under the top-level element
name may be used. </name>
```

Appendix D: EDM Element Dropdown Menu

```
<optgroup><option value="0">Select </option>
      <optgroup id="PCHO" label="edm:ProvidedCHO">
           <option value="1">owl:sameAs</option>
           <option value="2">dc:contributor</option>
           <option value="3">dc:coverage</option>
           <option value="4">dc:creator</option>
           <option value="5">dc:date</option>
           <option value="6">dc:description</option>
           <option value="7">dc:format</option>
           <option value="8">dc:identifier</option>
           <option value="9">dc:language</option>
           <option value="10">dc:publisher</option>
           <option value="11">dc:relation</option>
           <option value="12">dc:rights</option>
           <option>dc:source</option>
           <option>dc:subject</option>
           <option>dc:title</option>
           <option>dc:type</option>
           <option>dcterms:alternative</option>
           <option>dcterms:conformsTo</option>
           <option>dcterms:created</option>
           <option>dcterms:extent</option>
           <option>dcterms:hasFormat</option>
           <option>dcterms:hasPart</option>
           <option>dcterms:hasVersion</option>
           <option>dcterms:isFormatOf</option>
           <option>dcterms:isPartOf</option>
           <option>dcterms:isReferencedBy</option>
           <option>dcterms:isReplacedBy</option>
           <option>dcterms:isRequiredBy</option>
           <option>dcterms:issued</option>
           <option>dcterms:isVersionOf</option>
           <option>dcterms:medium</option>
           <option>dcterms:provenance</option>
           <option>dcterms:references</option>
           <option>dcterms:replaces</option>
           <option>dcterms:requires</option>
           <option>dcterms:spatial</option>
           <option>dcterms:tableOfContents</option>
           <option>dcterms:temporal</option>
           <option>edm:type</option>
           <option>edm:currentLocation</option>
           <option>edm:isNextInSequence</option>
      </optgroup>
      <optgroup id="WebR" label="edm:WebResource">
           <option>dc:rights</option>
           <option>edm:rights</option>
      </optgroup>
      <optgroup id="Agg" label="ore:Aggregation">
           <option>edm:aggregatedCHO</option>
           <option>edm:dataProvider</option>
           <option>edm:hasView</option>
           <option>edm:isShownBy</option>
           <option>edm:isShownAt</option>
           <option>edm:object</option>
           <option>edm:provider</option>
           <option>dc:rights</option>
           <option>edm:rights</option>
      </optgroup></optgroup>
```

Appendix E: Javascript with EDM Element Definitions and Constraints

```
selectJS2.html
<script>
var textBlocks = [
  'Select an option',
  '<b>owl:sameAs: <em>Ref, Optional, Repeatable</em></b> Use to point to your own
(linked data) representation of the object, if you have already minted a URI identifier
for it. It is also possible to provide URIs minted by third-parties for the object.',

  '<b>dc:contributor: <em>Literal/Ref, Optional, Repeatable</em></b> For contributors
to the CHO. If possible supply the identifier of the contributor from an authority
source.',
  '<b>dc:overage: <em>Literal/Ref, Optional, Repeatable</em></b> The spatial or
temporal topic of the CHO. Use the more precise dcterms:spatial or dcterms:temporal
properties if the data will support it. One of dc:coverage or dc:subject or dc:type or
dcterms:spatial must be provided',
  '<b>dc:creator: <em>Literal/Ref, Optional, Repeatable</em></b> For the creator of the
CHO. If possible supply the identifier of the creator from an authority source.',
   '<b>dc:date: <em>Literal/Ref, Optional, Repeatable</em></b> Use for a significant
date in the life of the CHO. Consider the subproperties of dcterms:created or
dcterms:issued.',
    '<b>dc:description: <em>Literal/Ref, Optional, Repeatable</em></b> A description of
the CHO. Either dc:description or dc:title must be provided.',
       '<b>dc:format: <em>Literal/Ref, Optional, Repeatable</em></b> Use for the format
of a born digital cultural heritage objects or for other terms applied to the CHO',
       '<b>dc:identifier: <em>Literal, Optional, Repeatable</em></b> An identifier of
the original CHO.',
       '<b>dc:language: <em>Literal, Optional, Repeatable</em></b> The language of text
CHOs and also for other types of CHO if there is a language aspect. Mandatory for TEXT
objects, strongly recommended for other object types with a language element.',
       '<b>dc:publisher: <em>Literal/Ref, Optional, Repeatable</em></b> The name of the
publisher of the CHO. If possible supply the identifier of the publisher from an
authority source.',
       '<b>dc:relation: <em>Literal/Ref, Optional, Repeatable</em></b> The name or
identifier of a related resource, generally used for other related CHOs. Cf
edm:isRelatedTo',
       '<b>dc:rights: <em>Literal/Ref, Optional, Repeatable</em></b> Use to give the
name of the rights holder of the CHO if possible or for more general rights
information. Note the difference between this property and the use of the controlled
edm:rights property which relates to the digital objects (see WebResource and
Aggregation tables).',
       '<b>dc:source: <em>Literal/Ref, Optional, Repeatable</em></b> The source of the
original CHO. This property should no longer be used for the name of the content
holder: for this, see edm:dataProvider in the ore:Aggregation table below.',
       '<b>dc:subject: <em>Literal/Ref, Optional, Repeatable</em></b> The subject of
the CHO. One of dc:subject or dc:coverage or dc:type or dcterms:spatial must be
provided',
       '<b>dc:title: <em>Literal, Optional, Repeatable</em></b> The title of the CHO.
Either dc:title or dc:description must be provided.',
       '<b>dc:type: <em>Literal/Ref, Optional, Repeatable</em></b> The nature or genre
of the CHO. Ideally the term(s) will be taken from a controlled vocabulary. One of
dc:type or dc:subject or dc:coverage or dcterms:spatial must be provided',
       '<b>dcterms:alternative: <em>Literal, Optional, Repeatable</em></b> Any
alternative title of the CHO including abbreviations or translations.',
       '<b>dcterms:conformsTo: <em>Literal/Ref, Optional, Repeatable</em></b> An
established standard to which the CHO conforms.',
       '<b>dcterms:created: <em>Literal/Ref, Optional, Repeatable</em></b> The date of
creation of the CHO.',
       '<b>dcterms:extent: <em>Literal/Ref, Optional, Repeatable</em></b> The size or
duration of the CHO.',
       '<b>dcterms:hasFormat: <em>Literal/Ref, Optional, Repeatable</em></b> A resource
related to the CHO that is substantially the same as the CHO but in another format.',
```

```
	'<b>dcterms:hasPart: <em>Literal/Ref, Optional, Repeatable</em></b> A resource
that is included either physically or logically in the CHO.',
	'<b>dcterms:hasVersion: <em>Literal/Ref, Optional, Repeatable</em></b> Another
resource that is a version, edition or adaptation of the CHO demonstrating substantive
changes in content rather than format.',
	'<b>dcterms:isFormatOf: <em>Literal/Ref, Optional, Repeatable</em></b> Another
resource that is substantially the same as the CHO but in another format.',
	'<b>dcterms:isPartOf: <em>Literal/Ref, Optional, Repeatable</em></b> A resource
in which the CHO is physically or logically included. This property can be used for
objects that are part of a hierarchy and will be used to support an appropriate display
in the portal. For that purpose it will be necessary to supply an identifier as the
value.',
	'<b>dcterms:isReferenceBy: <em>Literal/Ref, Optional, Repeatable</em></b>
Another resource that references, cites or otherwise points to the CHO.',
	'<b>dcterms:isReplacedBy: <em>Literal/Ref, Optional, Repeatable</em></b> Another
resource that supplants , displaces, or supersedes the CHO.',
	'<b>dcterms:isRequiredBy: <em>Literal/Ref, Optional, Repeatable</em></b> Another
related resource that requires the CHO to support its function, delivery or
coherence.',
	'<b>dcterms:issued: <em>Literal/Ref, Optional, Repeatable</em></b> Date of
formal issuance or publication of the CHO.',
	'<b>dcterms:isVersionOf: <em>Literal/Ref, Optional, Repeatable</em></b> Another
resource of which the CHO is a version, edition or adaptation, demonstrating
substantive changes in content rather than format.',
	'<b>dcterms:medium: <em>Literal/Ref, Optional, Repeatable</em></b> The material
or physical carrier of the CHO.',
	'<b>dcterms:provenance: <em>Literal/Ref, Optional, Repeatable</em></b> statement
of changes in ownership and custody of the CHO since its creation. Significant for
authenticity, integrity and interpretation.',
	'<b>dcterms:references: <em>Literal/Ref, Optional, Repeatable</em></b> Other
resources referenced, cited or otherwise pointed to by the CHO.',
	'<b>dcterms:replaces: <em>Literal/Ref, Optional, Repeatable</em></b> A related
resource that is supplanted, displaced, or superseded by the CHO.',
	'<b>dcterms:requires: <em>Literal/Ref, Optional, Repeatable</em></b> Another
resource that is required by the described resource to support its function, delivery
or coherence.',
	'<b>dcterms:spatial: <em>Literal/Ref, Optional, Repeatable</em></b> Spatial
characteristics of the CHO. i.e. what the CHO represents or depicts in terms of space
(e.g. a location, coordinate or place). Either dcterms:spatial or dc:type or dc:subject
or dc:coverage must be provided',
	'<b>dcterms:tableOfontents: <em>Literal, Optional, Repeatable</em></b> A list of
subunits of the CHO.',
	'<b>dcterms:temporal: <em>Literal/Ref, Optional, Repeatable</em></b> Temporal
characteristics of the CHO. i.e. what the CHO is about or depicts in terms of time
(e.g. a period, date or date range.)',
	'<b>edm:type: <em>Literal, Mandatory, Nonrepeatable</em></b> The provided object
is one of the types accepted by Europeana and will govern which facet it appears under
in the portal - TEXT, VIDEO, SOUND, IMAGE, 3D <br /><input type="radio" name="type"
value="text"/> Text<br /><input type="radio" name="type" value="video"/> Video<br
/><input type="radio" name="type" value="sound"/> Sound<br /><input type="radio"
name="type" value="image"/> Image<br /><input type="radio" name="type" value="3d"/>
3D<br />',

	'<b>edm:currentLocation: <em>Ref, Optional, Nonrepeatable</em></b> The
geographic location whose boundaries presently include the CHO. If the name of a
repository, building, site, or other entity is used then it should include an
indication of its geographic location.',
	'<b>edm:isNextInSequence: <em>Ref, Optional, Nonrepeatable</em></b> The
identifier of the preceding object where both objects are part of the same overall
resource. Use this for objects that are part of a hierarchy or sequence to ensure
correct display in the portal.',
	'<b>dc:rights: <em>Literal/Ref, Optional, Repeatable</em></b> Use for the name
of the rights holder of this digital representation if possible or for more general
rights information. Note the difference between this property and the use of the
mandatory, controlled edm:rights property below.',
```

```
       '<b>edm:rights: <em>Literal/Ref, Optional, Nonrepeatable</em></b> The value in
this element will indicate the usage and access rights that apply to this digital
representation. For the first implementation of EDM only the rights associated with the
ore:Aggregation class will be implemented. Later implementations will be able to use
the edm:rights associated with individual WebResources so it is strongly recommended
that a value is supplied for this property for each instance of a WebResource. <em>See
also edm:rights in the ore:Aggregation table.</em>',
       '<b>edm:aggregatedCHO: <em>Ref, Mandatory, Nonrepeatable</em></b> The identifier
of the source object e.g. the Mona Lisa itself. This could be a full linked open data
URI or an internal identifier.',
       '<b>edm:dataProvider: <em>Literal/Ref, Mandatory, Nonrepeatable</em></b> The
name or identifier of the data provider of the object (i.e. the organisation providing
data to an aggregator). Identifiers will not be available until Europeana has
implemented its Organisation profile.',
       '<b>edm:hasView: <em>Ref, Optional, Repeatable</em></b> The URL of a web
resource which is a digital representation of the CHO. This may be the source object
itself in the case of a born digital cultural heritage object. edm:hasView should only
be used where there are several views of the CHO and one (or both) of the mandatory
edm:isShownAt or edm:isShownBy properties have already been used. It is for cases where
one CHO has several views of the same object. (e.g. a shoe and a detail of the label of
the shoe)',
       '<b>edm:isShownBy: <em>Ref, Optional, Nonrepeatable</em></b> The URL of a web
view of the object. Either edm:isShownAt or edm:isShownBy is mandatory. For the rights
that will apply to previews please see edm:rights below.',
       '<b>edm:isShownAt: <em>Ref, Optional, Nonrepeatable</em></b> The URL of a web
view of the object in full information context. Either edm:isShownAt or edm:isShownBy
is mandatory. For the rights that will apply to previews please see edm:rights below.',
       '<b>edm:object: <em>Ref, Optional, Nonepeatable</em></b>The URL of a
representation of the CHO which will be used for generating previews for use in the
Europeana portal. This may be the same URL as edm:isShownBy. See <a
href="http://version1.europeana.eu/web/guest/technical-requirements/">Europeana Portal
Image Guidelines</a> for information regarding the specifications of previews ',
       '<b>edm:provider: <em>Literal/Ref, Mandatory, Nonrepeatable</em></b> The name or
identifier of the provider of the object (i.e. the organisation providing data directly
to Europeana). Identifiers will not be available until Europeana has implemented its
Organisation profile.',
       '<b>dc:rights: <em>Literal, Optional, Repeatable</em></b> Ideally this should be
applied to the edm:WebResource or the edm:ProvidedCHO. It is included here for the
conversion of data from ESE where it is not known which object the rights apply to.',
       '<b>edm:rights: <em>Ref, Mandatory, Nonrepeatable</em></b> This is a mandatory
property and the value given here should be the rights statement that applies to the
digital representation at the URL given in edm:object or edm:isShownAt/By. The value
should be taken from one of those listed in the <a
href="http://version1.europeana.eu/web/guest/technical-requirements/">Europeana Rights
Guidelines</a> The rights statement given in this property will also apply to the
previews used in the portal and will be the source of: (1) the entry in the Rights
facet in the portal, (2) the license badge that appears under the preview on the result
page.  Where there are several web resources attached to one edm:ProvidedCHO the rights
statement given here will be regarded as the "reference" value for all the web
resources so a suitable value should be chosen if the rights statements vary between
different resources. In future implementations it is hoped to handle rights statements
for separate web resources associated with one CHO separately.',
];

function changeText(elemid, displayId) {
  var ind = document.getElementById(elemid).selectedIndex;
  document.getElementById(displayId).innerHTML = textBlocks[ind];
}
</script>
```

Appendix F: Cognitive Walkthrough

Task	Will the user try to achieve the right effect?	Will the user notice that the correct action is available?	Will the user associate the correct action with the effect they are trying to achieve?	Will the user see that progress is being made toward the solution of their task?
One-to-One				
The user must first identify the row for the mods:title element	Yes. Given that the user is attempting to make a mapping between their source metadata and EDM, they will anticipate making element-by-element mappings, and will therefore focus on one element at a time. This example has them focusing on the mods:title element.	Yes. The righthand columns all start out identical in each row, the two columns on the left are unique in each row. A user used to using tables will look to the top row for column headings and read columns left to right. Looking at the left most column, the user will be able to identify that each row represents a separate element, and be able to locate the row for the mods:title element	Yes. The user will see the element name mods:title, and realize that this is the information concerning the source metadata and where they should start when attempting to map to the target schema.	Yes. When the user identifies the mods:title element row, they will realize realize they are on the right path, and are now able to move toward the next step of the mapping
Read the information about the source metadata and the element description.	Yes. After identifying the correct row for the element they are trying to map, the user may want to know more precise information about the use of the element field, and would likely want to consult the element definition prior to making a mapping decision.	Yes. The element description is directly to the right of the element name making it very visible for the user. It follows a logical left-to-right eye movement.	Yes. Again, the definition for the source element appears directly to the right of the element name and instance information. Given that all of the elements are in a single, vertical column, the user will view the contents of this row as pertinent to the mods:title source element.	Yes. As the user is aware of the purpose of mapping the source schema to the target schema, they will understand that the definition of the element can be an important aid tot eh mapping. Reading through the official definition, they may be able gain a greater understanding of any nuances of the element use.
Click the dropdown menu	Yes. The user is making a mapping between two metadata schema, and will know that they must make element mappings. After looking over the information for a source element, they will next want to look at the potential target elements available to find the best match.	Yes. The dropdown menu is direclty to the right of the element definition, and as the user continues to follow the left-to-right eye motion, they will see the dropdown menu for the row and along with its default value ("Select an EDM element") and realize this is how they proceed with the mapping. The column heading will also give an indication that this is where the EDM element is chosen.	Yes. The user will associate this row's dropdown menu with mods:title. The dropdown menu is right next to the element definition and is a logical next step. A dropdown menu in HTML is associated with selecting a choice, making a decision; the user will see this is where their input comes in.	Yes. When the user clicks on the dropdown box, all of the EDM elements will appear in the menu to select from.
Reading through the available elements, dc:title is chosen for its syntactic similarity to the source element name	Yes. For an element like mods:title, which is relatively straight forward, the user is likely to scan through the available target elements to find one that looks like it would contain a title.	Yes. After clicking into the dropdown menu, the user will infer that they are meant to make a choice of one of the options below. Looking down the list, the user will see dc:title as an option.	Yes. Seeing dc:title, the user would select it, assuming there would be a significant similarity between it and mods:title. However, if there is any uncertainty, the user may not initially know that the tool will display the definition of the selected element. While many of the element names provide some guidance as to what values they will contain, detailed definitions are really required to make an informed decision on element mappings. They may not realize that the definition will appear as an aid to making the desicion and believe that they must make the mapping based on the target element names alone. However, if this is the case, they would still likely select dc:title for its syntactic similarity to mods:title.	Yes. After dc:title is selected, the element definition will appear to the right of the dropdown menu. Showing that their selection was recognized.
***When dc:title is selected, the element description will appear next to the dropdown menu.* Compare this definition to the definition for mods:title**	Yes. After making a preliminary choice for the target element to be mapped to mods:title, the user will want to ensure that the definition, allowable values, etc. align with those of the source element.	Yes. Once the dc:title element is selected, the definition appears next to the dropdown menu. The change of information on the screen will draw the user's eye, and the wording, as well as the column heading ("EDM Element Descriptions") will inform the user that this is the target element definition.	Yes. Once the user recognizes the target element definition for what it is, they will realize they should read it through, ensuring that the source element metadata would appropriately fit within the definition and explanation of the target element.	Yes. As the user compares the source element informaiton and definition to that of the target element, they will begin to verify the semantic similarities expected based upon the syntactic similarities of the element names.
seeing that they are semantically equal, decide upon this mapping	Yes. Mods:title and dc:title are semantically similar. As the user reads through the definition and provided information for each the source and target elements, it is more than likely that these similarities will be viewed as adequate justification for creating an equivalent mapping between the elements.	Yes. The user will realize that after selecting a potential source element, they will need to decide whether or not this is an appropriate mapping.	Yes. The user wants to make an appropriate mapping from the target element to the source schema. They want the tool to display the mapping that they decide upon. They want to make an appropriate mapping decision for this element, so they are then able to move on to the subsequent elements to continue the mapping. In order to ensure that this mapping is adequate, they will want to review all of the information that the tool provides on each of the elements to ensure there is no problem with the selected mapping.	Yes. As the user makes a final descision on this mapping, they will realize they can move on to the mapping of source element in the next row to further complete the full metadata mapping.

Content Requirement - Controlled Vocab				
Read the required Edm element list at the top of the screen.	Yes. The user will want to read information that will aid them in completing the mapping, especially with regards to required target elements to ensure a complete mapping.	Yes. The information is at the top of the screen. It is highly visible, and should be easily seen by the user when the mapping tool screen loads.	Yes. Even a cursory glance at the information at the top of the screen will show the user that this is the information pertaining to required EDM elements, and will aid them in the completion of their mapping.	Yes. The user will be ensuring they have a good understanding of the required elements of EDM, which is progress towards the completion of the mapping of the source metadata to EDM.
Realize there is no equivalent to edm:type in source metadata.	No. After seeing the small number of required fields, the user would likely want to ensure that each of these fields was completed. Either after completing all of the source element mappings or prior to beginning any other mappings. The user will want to check whether or not there is an appropriate source element for each of the required fields, and if not, need to device a solution for filling these fields.	Yes. The information to determine whether or not there is a source element appropriate for edm:type is throughout the mapping tool. Each source element has metadata information and a definition, which can be compared individually to the edm:type description to determine there is no adequate source element.	Yes. Comparing the required edm:type's element description to those of the source elements is the clear way to determine whether or not there is any adequate mapping to edm:type from the source metadata.	Yes. With the realization that there is no adequate source element to be mapped to edm:type, the user will realize they must provide the values for edm:type in another way.
See note for additional rows at bottom of table.	Yes. The user will be looking for a way to specify the contents for edm:type separate from the main mapping tool which is based on mapping source elements.	Yes. The note element is listed below the informaiton for the required EDM elements as a resolution for the issue of no adequate source element.	Yes. As the user is looking for a way to specify edm:type, they will see that the note for additional rows at the bottom of the table is meant as a resolution for unmatched mandatory target elements.	Yes. The user will see that this is an alternative to mapping source elements, and a possible resolution to the unmatched edm:type element.
Scroll to bottom of table.	Yes. The note the user read directed them to the bottom of the table, so they would follow the note's instructions and scroll down.	Yes. The user is familiar with browser functionality, and will know that they are able to scroll down to the bottom of a browser page.	Yes. The note informed the user that unmatched mandatory elements can be filled out at the bottom of the page, and the user is aware of how to get to the bottom of the page, and will therefore scroll down in the browser.	Yes. The user will realize they are following the instructions of the mapping tool, and as they get to the bottom of the table, they will see the additional rows set aside for the specification for unmatched mandatory elements.
Click first free dropdown menu.	Yes. The user wants to choose the element they want to specify.	Yes. Having been directed to the bottom of the table, the user will see the available dropdown menus for the selection of EDM elements.	Yes. Again, the information at the top of the page directed the user to the bottom of the table to specify unmatched mandatory EDM elements, and having followed the instructions to the bottom of the page, will know that this is where they are supposed to specify the required edm:type.	Yes. When the menu is clicked, it drops down, showing the user all of the possible edm elements to choose from.
Select edm:type.	Yes. The user knows they must specify the required edm:type, and this is the element they are trying to specify now.	Yes. After clicking the dropdown menu, they will see selection options which must be chosen from, and one of them is edm:type.	Yes. The user knows they want to specify edm:type and sees that this is one of the option and would therefore select edm:type from the dropdown.	Yes. After edm:type is selected, the definition, along with the five possible values appear (with radial buttons alongside).
***Radial buttons appear.* Look at source metadata. Realize the collection is all images.**	When the five options appear, the user knows they must figure out which one adequate describes the objects which the source metadata is describing. To do this they must look at the metadata. However, it is likely that they may already know enough about the collection to know that all of the records are for images, and will therefore not need to look at the source metadata at all.	Yes, the information for the source metadata fills the left side of the table and is available for them to peruse. They may have to look externally for this information, however.	Yes. Having even only glanced at the table, they will see that information about the source metadata is available within the table. If there is not enough information here to ascertain that this is an image collection, they will try to discover the answer in other ways. Perhaps through previous knowledge of the collection at hand, returning to the OAI-PMH sets description, or to the original context of the collection.	Yes. Making the realization that the records are an image collection, they will know that they
Select IMAGE option.	Yes. Once the user is sure that the collection is of images, the user will want to specify this in the edm:type radial options.	Yes. They already saw the five options appear, and will realize they must choose one.	Yes. They scrolled to the bottom of the page to make this required specification for edm:type, and when the radial buttons appeared would realize that they must choose one of these. Once they decide that the entire collection is of images, they will return to these radials, and seeing IMAGE as a choice, will select it.	Yes. Once the button next to IMAGE is selected, the user will see that they have specified the edm:type element for the collection, as required.

Unmatched Mandatory Target Elements				
Read the required EDM element list at the top of the screen.	Yes. The user will want to read information that will aid them in completing the mapping, especially with regards to required target elements to ensure a complete mapping.	Yes. The information is at the top of the screen. It is highly visible, and should be easily seen by the user when the mapping tool screen loads.	Yes. Even a cursory glance at the information at the top of the screen will show the user that this is the information pertaining to required EDM elements, and will aid them in the completion of their mapping.	Yes. The user will be ensuring they have a good understanding of the required elements of EDM, which is progress towards the completion of the mapping of the source metadata to EDM.
Realize there is no equivalent to edm:DataProvider in source metadata.	No. After seeing the small number of required fields, the user would likely want to ensure that each of these fields was completed. Either after completing all of the source element mappings or prior to beginning any other mappings. The user will want to check whether or not there is an appropriate source element for each of the required fields, and if not, need to device a solution for filling these fields.	Yes. The information to determine whether or not there is a source element appropriate for edm:type is throughout the mapping tool. Each source element has metadata information and a definition, which can be compared individually to the edm:type description to determine there is no adequate source element.	Yes. Comparing the required edm:dataProvider's element description to those of the source elements is the clear way to determine whether or not there is any adequate mapping to edm:type from the source metadata.	Yes. With the realization that there is no adequate source element to be mapped to edm:dataProvider, the user will realize they must provide the values for edm:type in another way.
See note for additional rows at bottom of table.	Yes. The user will be looking for a way to specify the contents for edm:dataProvider separate from the main mapping tool which is based on mapping source elements.	Yes. The note element is listed below the informaiton for the required EDM elements as a resolution for the issue of no adequate source element.	Yes. As the user is looking for a way to specify edm:dataProvider, they will see that the note for additional rows at the bottom of the table is meant as a resolution for unmatched mandatory target elements.	Yes. The user will see that this is an alternative to mapping source elements, and a possible resolution to the unmatched edm:dataProvider element.
Scroll to bottom of table.	Yes. The note the user read directed them to the bottom of the table, so they would follow the note's instructions and scroll down.	Yes. The user is familiar with browser functionality, and will know that they are able to scroll down to the bottom of a browser page.	Yes. The note informed the user that unmatched mandatory elements can be filled out at the bottom of the page, and the user is aware of how to get to the bottom of the page, and will therefore scroll down in the browser.	Yes. The user will realize they are following the instructions of the mapping tool, and as they get to the bottom of the table, they will see the additional rows set aside for the specification for unmatched mandatory elements.
Click first free dropdown menu.	Yes. The user wants to choose the element they want to specify.	Yes. Having been directed to the bottom of the table, the user will see the available dropdown menus for the selection of EDM elements.	Yes. Again, the information at the top of the page directed the user to the bottom of the table to specify unmatched mandatory EDM elements, and having followed the instructions to the bottom of the page, will know that this is where they are supposed to specify the required edm:type.	Yes. When the menu is clicked, it drops down, showing the user all of the possible edm elements to choose from.
Select edm:dataProvider.	Yes. The user knows they must specify the required edm:dataProvider, and this is the element they are trying to specify now.	Yes. After clicking the dropdown menu, they will see selection options which must be chosen from, and one of them is edm:dataProvider.	Yes. The user knows they want to specify edm:dataProvider and sees that this is one of the option and would therefore select edm:type from the dropdown.	Yes. After edm:dataProvider is selected, the element definition appears.
***the element description appears next to the dropdown menu.* Write the provider of the metadata in the value field.**	Yes. Since there is no source element to match edm:dataProvider, there are no values for the element. Therefore, they know they will have to specify what the element is. After reading the definition of the element, they will know that the correct value for the element for all of the records is the organization that is providing the data, which it is reasonable to assume that the user would know.	Yes. The value field is right next to the dropdown menu and EDM element definition that appeared when edm:dataProvider was chosen. The field is a text box, which, as the user is familiar with browsers, is an obvious place for them to enter information.	Yes. The user knows that they must specify the contents of the edm:dataProvider element, and seeing the text value field right next to the edm:dataProvider information, they will assume that this is where they must enter in the information about the data provider.	Yes. Once the user types in the data provider information, the mapping tool will clearly show that the edm:dataProvider element should include this value.

One-to-Many Identical Target Elements				
First, the user must identify the row for the subject element.	Yes. Given that the user is attempting to make a mapping between their source metadata and EDM, they will anticipate making element-by-element mappings, and will therefore focus on one element at a time. This example has them focusing on the mods:title element.	Yes. The righthand columns all start out identical in each row, the two columns on the left are unique in each row. A user used to using tables will look to the top row for column headings and read columns left to right. Looking at the left most column, the user will be able to identify that each row represents a separate element, and be able to locate the row for the mods:title element	Yes. The user will see the element name mods:title, and realize that this is the information concerning the source metadata and where they should start when attempting to map to the target schema.	Yes. When the user identifies the mods:title element row, they will realize realize they are on the right path, and are now able to move toward the next step of the mapping
Reading the information about the source metadata, the user sees the example values, which show that all of each object's subjects are in a single occurrence of the field.	No. While the sample values are there for the user to look at, the user may not look very closely at them, and may not realize that the subject keywords are actually within the same field simply from their first look at the information. They may not realize there are multiple subjects in a single occurence of the elementwithout really looking closely at the source metadata inforamtion.	Yes. The User is likely to see all the information from the source metadata within the subject element cell, including the sample values which are within the cell and show that there are multiple subjects in a single field occurrence	No. The user is not likely to realize that the record subjects are contained with the same occurrence of the element just from looking at the sample values presetned by the mapping tool.	Yes. If the user does read through the sample values and is able to discern that all of the subjects are in a sincle occurrence of the subject element, they will see progress. Though they will realize this complicates the task, having realized the complication is there, they will be able to consider a solution.
The user reads the definition of the subject element.	Yes. After identifying the correct row for the element they are trying to map, the user may want to know more precise information about the use of the element field, and would likely want to consult the element definition prior to making a mapping decision.	Yes. The element description is directly to the right of the element name making it very visible for the user. It follows a logical left-to-right eye movement.	Yes. Again, the definition for the source element appears directly to the right of the element name and instance information. Given that all of the elements are in a single, vertical column, the user will view the contents of this row as pertinent to the mods:title source element.	Yes. As the user is aware of the purpose of mapping the source schema to the target schema, they will understand that the definition of the element can be an important aid tot eh mapping. Reading through the official definition, they may be able gain a greater understanding of any nuances of the element use.
Click the dropdown menu.	Yes. The user is making a mapping between two metadata schema, and will know that they must make element mappings. After looking over the information for a source element, they will next want to look at the potential target elements available to find the best match.	Yes. The dropdown menu is direclty to the right of the element definition, and as the user continues to follow the left-to-right eye motion, they will see the dropdown menu for the row and along with its default value ("Select an EDM element") and realize this is how they proceed with the mapping. The column heading will also give an indication that this is where the EDM element is chosen.	Yes. The user will associate this row's dropdown menu with mods:title. The dropdown menu is right next to the element definition and is a logical next step. A dropdown menu in HTML is associated with selecting a choice, making a decision; the user will see this is where their input comes in.	Yes. When the user clicks on the dropdown box, all of the EDM elements will appear in the menu to select from.
Read through the available elements, and chose dc:subject for its syntactic similarity to the source element name.	Yes. For an element with the name 'subject', which is relatively straight forward, the user is likely to scan through the available target elements to find one that looks like it would contain a subject.	Yes. After clicking into the dropdown menu, the user will infer that they are meant to make a choice of one of the options below. Looking down the list, the user will see dc:subject as an option.	Yes. Seeing dc:subject, the user would select it, assuming there would be a significant similarity between it and mods:title. However, if there is any uncertainty, the user may not initially know that the tool will display the definition of the selected element. While many of the element names provide some guidance as to what values they will contain, detailed definitions are really required to make an informed decision on element mappings. They may not realize that the definition will appear as an aid to making the desicion and believe that they must make the mapping based on the target element names alone. However, if this is the case, they would still likely select dc:subject for its syntactic similarity to the source metadata element.	Yes. After dc:subject is selected, the element definition will appear to the right of the dropdown menu. Showing that their selection was recognized.
***When dc:subject is selected, the element definition and information appears next to the dropdown menu.* The user compares this definition to the definition for the source element subject.**	Yes. After making a preliminary choice of the target element to be mapped to subject, the user will want to ensure that the definition, allowable values, etc. align with those of the source element.	Yes. Once the dc:subject element is selected, the definition appears next to the dropdown menu. The change of information on the screen will draw the user's eye, and the wording, as well as the column heading ("EDM Element Descriptions") will inform the user that this is the target element definition.	Yes. Once the user recognizes the target element definition for what it is, they will realize they should read it through, ensuring that the source element metadata would appropriately fit within the definition and explanation of the target element.	Yes. As the user compares the source element informaiton and definition to that of the target element, they will begin to verify the semantic similarities expected based upon the syntactic similarities of the element names.

The user sees that the definitions are semantically similar	Yes. As the user reads through the definition and provided information for each the source and target elements, it is more than likely that these similarities will be viewed as adequate justification for creating an equivalent mapping between the elements.	Yes. The user will realize that after selecting a potential source element, they will need to decide whether or not this is an appropriate mapping.	Yes. The user wants to make an appropriate mapping from the target element to the source schema. They want the tool to display the mapping that they decide upon. They want to make an appropriate mapping decision for this element, so they are then able to move on to the subsequent elements to continue the mapping. In order to ensure that this mapping is adequate, they will want to review all of the information that the tool provides on each of the elements to ensure there is no problem with the selected mapping.	Yes. As the user makes a final descision on this mapping, they will realize they can move on to the mapping of source element in the next row to further complete the full metadata mapping.
The user sees that dc:subject is repeatable - realizes source element should be expanded into multiple occurrences for dc:subject.	No. It is not guaranteed that the user will infer from the fact that dc:subject is repeatable that the source element should be expanded into multiple occurrences of the dc:subject element.	Yes. The information is included at the beginning of the definition of dc:subject that appears next to the dropdown menu.	No. While the user is likely to see that dc:subject is repeatable when reading through the element definition, they may not make the correlation between this and the fact that all of the subjects are in a single source element occurrence.	Yes. If the user, seeing that dc:subject is repeatable, realizes that the source element could be expanded into multiple occurrences instead of a single occurrence, then they are likely to see that this is a more appropriate mapping.
The user sees the note field and puts in the note: "Create separate dc:subject elements for each semi-colon separated subject in the source element."	Yes. Knowing that the source subject has multiple subjects in a single field, and dc:subject is repeatable, the user will likely decide that these subjects should be split up into multiple occurrences of dc:subject.	Yes. The Notes field is highly visible next to the dropdown menu. The user is very likely to see it and know that they are able to include notes in the field.	No. The user may not realize that the way they are meant to specify that the values in the source element should be separated is by writing an explicity note in the Note field. They may expect something more concrete -- something that is machine readable, for instance. If they do realize this is where they are meant to specify such instructions, they will likely type in an adequately specific note to inform the mapping.	Yes. If the users realizes this is how they are meant to specify the separation of the subjects into unique dc:subject fields, having come to the end of the row, they will see that the specification for the subject element is complete, and they will have completed this part of the mapping.

One-to-Many Unique Target Elements				
First, the user must identify the row for the author/contributors field.	Yes. Given that the user is attempting to make a mapping between their source metadata and EDM, they will anticipate making element-by-element mappings, and will therefore focus on one element at a time.	Yes. The righthand columns all start out identical in each row, the two columns on the left are unique in each row. A user used to using tables will look to the top row for column headings and read columns left to right. Looking at the left most column, the user will be able to identify that each row represents a separate element, and be able to locate the row for the source element	Yes. The user will see the source element name, and realize that this is the information concerning the source metadata and where they should start when attempting to map to the target schema.	Yes. When the user identifies the mods:title element row, they will realize realize they are on the right path, and are now able to move toward the next step of the mapping
Reading the information about the source metadata, the user sees the example values, which show that the main author and the contributors are all in the same field.	No. While the sample values are there for the user to look at, the user may not look very closely at them, and may not realize that the author/contributors are in the same field. However, as there is a consistent que within the field ('contributors'), they may pick up on this and realize there are multiple persons listed in a single field.	Yes. The sample values are listed in the same cell at the element name. As such, the user is like to see these when first coming to the row.	No. The user may realize that the record author and contributors are contained with the same occurrence of the element just from looking at the sample values presented by the mapping tool.	Yes. If the user does read through the sample values and is able to discern that the main author is in the same occurrence of the element as the contributors, they will see this realization as progress. Though they will realize this complicates the task, having realized the complication is there, they will be able to consider a solution.
The user reads the definition of the source element.	Yes. After identifying the correct row for the element they are trying to map, the user may want to know more precise information about the use of the element field, and would likely want to consult the element definition prior to making a mapping decision.	Yes. The element description is directly to the right of the element name making it very visible for the user. It follows a logical left-to-right eye movement.	Yes. Again, the definition for the source element appears directly to the right of the element name and instance information. Given that all of the elements are in a single, vertical column, the user will view the contents of this row as pertinent to the mods:title source element.	Yes. As the user is aware of the purpose of mapping the source schema to the target schema, they will understand that the definition of the element can be an important aid tot eh mapping. Reading through the official definition, they may be able gain a greater understanding of any nuances of the element use.
Clicks the dropdown menu.	Yes. The user is making a mapping between two metadata schema, and will know that they must make element mappings. After looking over the information for a source element, they will next want to look at the potential target elements available to find the best match.	Yes. The dropdown menu is direclty to the right of the element definition, and as the user continues to follow the left-to-right eye motion, they will see the dropdown menu for the row and along with its default value ("Select an EDM element") and realize this is how they proceed with the mapping. The column heading will also give an indication that this is where the EDM element is chosen.	Yes. The user will associate this row's dropdown menu with mods:title. The dropdown menu is right next to the element definition and is a logical next step. A dropdown menu in HTML is associated with selecting a choice, making a decision; the user will see this is where their input comes in.	Yes. When the user clicks on the dropdown box, all of the EDM elements will appear in the menu to select from.

Reads through the available elements. Chooses the dc:contributor element.	No. Given that the word 'contributors' is in the element value, they might see that the dc:contribor option may be fitting for at least part of the contents of the element.	Yes. The dc:contributor option is listed in the dropdown menu. After the user clicks into the dropdown menu, they are likely to read down the options until they find a possible target element to map to.	Yes. The user is looking for potential elements to map to. Seeing that dc:contributor is similar to some of the contents of the source element value, the user would select it, assuming there would be a significant similarity. However, if there is any uncertainty, the user may not initially know that the tool will display the definition of the selected element. While many of the element names provide some guidance as to what values they will contain, detailed definitions are really required to make an informed decision on element mappings. They may not realize that the definition will appear as an aid to making the desicion and believe that they must make the mapping based on the target element names alone. However, if this is the case, they would still likely select dc:contributor at some point due to the nature of the value of the source element.	Yes. The definition fo dc.contributor will appear next to the dropdown menu and the user will see that this is a way for them to determine whether or not this is the appropriate mapping.
***Definition appears.* Compare to source element definition. Sees that this is appropriate for all but the author.**	Yes. The user will want to know if the definition of dc:contributor aligns at all with that of the source element. As such , they are likely to compare this to the source element definition and information	Yes. Both of the definitions are highly visible. The EDM element definition appeared when dc:contributor was selected, drawing the eye. When the user reads this, they are likely to want to compare this which provides the information on allowable values to the information they have about the source element to explore whether there is semantic alignment.	Yes. Once the user recognizes the target element definition for what it is, they will realize they should read it through, ensuring that the source element metadata would appropriately fit within the definition and explanation of the target element.	Yes. As the user compares the source element information and definition to that of the target element, they will begin to verify the semantic similarities between the contributor portion of the source element and the target element dc:contributor. They will see that this is a viable chioce at least for lpart of the element. However, not for the author at the beginning of the element value.
Clicks the dropdown menu again.	Yes. Seeing that the dc:contributor field does not completely cover the contents of the source element, the user will want to look at other potential target elements to map the source element to.	Yes. The user has already used the dropdown menu, and knows where it is. The user should still be focused on the same row, which is where the relative dropdown menu is located.	Yes. The user has already used the dropdown menu, and knows it contains the list of potential EDM elements from which to choose. They will know if there are any other possible EDM elements besides dc:contributor, that it will be present in the dropdown menu.	Yes. When the dropdown menu is clicked, it expands showing all the potential EDM elements.
Looks for author but does not find. Locates dc:creator field instead.	Yes. Seeing that dc:contributor fits well with all but the author portion of the source element value, the user will be lookig for an element that adequately represents this portion of the field.	Yes. The dc:creator option is listed in the dropdown menu. After the user clicks into the dropdown menu, they are likely to read down the options until they find a possible target element to map to.	Yes. 'creator' has some semantic relationship to 'author' which, the user will likely see as a potential target element to map to.	Yes. Again, the definition of dc:creator will appear when it is selected in the dropdown menu. The user will see that this is progress toward determining whether or not this ais an appropriate target element to map the source element to.
Compares to source element definition. Sees that this is appropriate for the author.	Yes. After making another choice for the target element to be mapped to the source element, the user will want to discover whether the definition, allowable values, etc. align with those of the source element.	Yes. Once the dc:creator element is selected, the definition appears next to the dropdown menu. The change of information on the screen will draw the user's eye, and the wording, as well as the column heading ("EDM Element Descriptions") will inform the user that this is the target element definition.	Yes. Once the definition appears, the user will again want to compare it to the definition and information they have available for the source element.	Yes. As the user compares the source element information and definition to that of the target element, they will see that dc:creator is the proper field for the source metadata suthor element.
Cannot choose two target elements for a single source element. Instead, chooses dc:creator, and writes in Note field: "dc:creator for content before 'contributor:'. Create separate dc:contributor elements for each semi-colon separated name after 'contributor:'."	No. Having read the definitions for both dc:creator and dc:contributor, the user will likely see that separate portions of the source element values should be placed in either target element. And to make a quality mapping, the contents will have to separated into unique elements in EDM.	Yes. The Notes field is highly visible next to the dropdown menu. The user is very likely to see it and know that they are able to include notes in the field.	No. The user may not realize that the way they are meant to specify that the values in the source element should be separated is by writing an explicity note in the Note field. They may expect something more concrete -- something that is machine readable, for instance. If they do realize this is where they are meant to specify such instructions, they will likely type in an adequately specific note to inform the mapping.	No. While the note will extrapolate the mapping, the user may find this solution wanting.

Many-to-One Combination				
First, the user identifies the row for givenName	Yes. Given that the user is attempting to make a mapping between their source metadata and EDM, they will anticipate making element-by-element mappings, and will therefore focus on one element at a time. This example has them focusing on the mods:title element.	Yes. The righthand columns all start out identical in each row, the two columns on the left are unique in each row. A user used to using tables will look to the top row for column headings and read columns left to right. Looking at the left most column, the user will be able to identify that each row represents a separate element, and be able to locate the row for the givenName element	Yes. The user will see the element name givenName, and realize that this is the information concerning the source metadata and where they should start when attempting to map to the target schema.	Yes. When the user identifies the givenName element row, they will realize realize they are on the right path, and are now able to move toward the next step of the mapping
Looks at information from source metadata, sees sample values are only first names.	No. While the sample values are there for the user to look at, the user may not look very closely at them, and may not realize that there are only first names in the sample values. However, given the name of the element 'givenName' the user may ascertain this element only contains the first name of the author.	Yes. The sample values are listed in the same cell at the element name. As such, the user is like to see these when first coming to the row.	No. The user may realize that the element only includes the first name, and not the full name, of the author.	Yes. If the user does read through the sample values and is able to discern that the element only contains the first name of the author, they will see this realization as progress. Though they will realize this complicates the task, having realized the complication is there, they will be able to consider a solution.
Read definition	Yes. After identifying the correct row for the element they are trying to map, the user may want to know more precise information about the use of the element field, and would likely want to consult the element definition prior to making a mapping decision.	Yes. The element description is directly to the right of the element name making it very visible for the user. It follows a logical left-to-right eye movement.	Yes. Again, the definition for the source element appears directly to the right of the element name and instance information. Given that all of the elements are in a single, vertical column, the user will view the contents of this row as pertinent to the mods:title source element.	Yes. As the user is aware of the purpose of mapping the source schema to the target schema, they will understand that the definition of the element can be an important aid tot eh mapping. Reading through the official definition, they may be able gain a greater understanding of any nuances of the element use.
Definition says element includes the given name of the author and the last name of the author is in the element lastName.	Yes. After identifying the correct row for the element they are trying to map, the user may want to know more precise information about the use of the element field, and would likely want to consult the element definition prior to making a mapping decision.	Yes. The element description is directly to the right of the element name making it very visible for the user. It follows a logical left-to-right eye movement. The information about the element's relationship to the lastName element is included within this definition.	Yes. Again, the definition for the source element appears directly to the right of the element name and instance information. Given that all of the elements are in a single, vertical column, the user will view the contents of this row as pertinent to the mods:title source element.	Yes. As the user is aware of the purpose of mapping the source schema to the target schema, they will understand that the definition of the element can be an important aid tot eh mapping. Reading through the official definition, they may be able gain a greater understanding of any nuances of the element use.
Locates the row for lastName.	No. Given that it is mentioned in conjunction with the source element the user is attempting to map, they may wish to look at both rows simultaneously to consider the appropriate mapping.	Yes. The righthand columns all start out identical in each row, the two columns on the left are unique in each row. A user used to using tables will look to the top row for column headings and read columns left to right. Looking at the left most column, the user will be able to identify that each row represents a separate element, and be able to locate the row for the lastName element	Yes. The user will see the element name lastName, and realize that this is the information concerning the source metadata and where they should start when attempting to map to the target schema.	Yes. When the user identifies the lastName element row, they will realize realize they are on the right path, and are now able to move toward the next step of the mapping
Looks at information from source metadata, sees sample values are last names.	Yes. As the user came to this row from the information found in the givenName element definition, they will want to examine the information here in the lastName row to ascertain its correspondence with the givenName element, particularly that the sample values are indeed surnames.	Yes. The sample values are listed in the same cell at the element name. As such, the user is like to see these when first coming to the row.	Yes. Having come to this element row on the specific task to see how it relates to that of the givenName element they are attempting to map, they will likely want to look at all the information and will see that the information in the cell with the lastName element name will provide more detailed information pertaining to this field.	Yes. The user will confirm that this element supplies the surname of the author to be paired with the first name provided in the givenName element.
Reads definition. Sees it compliments givenName definition.	Yes. After identifying the correct row for lastName, the user will want to compare this definition to that of the givenName element.	Yes. The element description is directly to the right of the element name making it very visible for the user. It follows a logical left-to-right eye movement.	Yes. Again, the definition for the source element appears directly to the right of the element name and instance information. Given that all of the elements are in a single, vertical column, the user will view the contents of this row as pertinent to the mods:title source element.	Yes. As the user is aware of the purpose of mapping the source schema to the target schema, they will understand that the definition of the element can be an important aid to the mapping. Reading through the official definition, they will be able to see it is the last name of the author, complimenting the givenName element.
Returns to givenName.	No. The user was initially attempting to complete the mapping for givenName, so they may return to it. Yet, having to move back and forth between two source metadata elements might cause confusion.	Yes. Having already located the row for givenName, the user will know it is there already.	Yes. If the user wants to complete the mapping for givenName, they will know to return to the element row in the mapping tool.	No. The user may be confused and unsure about how to proceed when they are considering two separate elements in two separate rows.

Clicks the dropdown menu.	Yes. The user is making a mapping between two metadata schema, and will know that they must make element mappings. After looking over the information for a source element, they will next want to look at the potential target elements available to find the best match.	Yes. The dropdown menu is directly to the right of the element definition, and as the user continues to follow the left-to-right eye motion, they will see the dropdown menu for the row and along with its default value ("Select an EDM element") and realize this is how they proceed with the mapping. The column heading will also give an indication that this is where the EDM element is chosen.	Yes. The user will associate this row's dropdown menu with mods:title. The dropdown menu is right next to the element definition and is a logical next step. A dropdown menu in HTML is associated with selecting a choice, making a decision; the user will see this is where their input comes in.	Yes. When the user clicks on the dropdown box, all of the EDM elements will appear in the menu to select from.
Read through the available elements. Sees there is no givenName equivalent, nor is there an author element.	Yes. When the dropdown options appear the user will understand that they must select an option for the mapping of the source element. They will look for elements that might match with givenName, or as they know from the definition this is a part of the author name, an author element or equivalent.	Yes. When the menu is clicked, all of the possible EDM choices dropdown. The user will understand they must select one.	Yes. They clicked on the dropdown menu looking for the options for mapping the source element to EDM, and will therefore look to the dropdown menu contents in order to continue with their mapping.	Yes. Having failed to find syntactically similar EDM elements for the source element, the user will see that they must again at the EDM elements for names that may also matched well with the contents of the source element.
Locates and selects dc:creator element.	Yes. The user will be looking for an appropriate element to map to, and having checked for syntactically similar element names, will likely move on to element names that might be semantically similar.	Yes. The dc:creator option is listed in the dropdown menu. After the user clicks into the dropdown menu, they are likely to read down the options until they find a possible target element to map to. And once they see there is no givenName or author element, will look for other names that may share semantica similirities.	Yes. 'creator' has some semantic relationship to 'author' which, the user will likely see as a potential target element to map to.	Yes. Again, the definition of dc:creator will appear when it is selected in the dropdown menu. The user will see that this is progress toward determining whether or not this ais an appropriate target element to map the source element to.
***Definition appears.* Reads through the definition.**	Yes. After making a preliminary choice for the target element to be mapped to, the user will want to ensure that the definition, allowable values, etc. align with those of the source element.	Yes. Once the dc:creator element is selected, the definition appears next to the dropdown menu. The change of information on the screen will draw the user's eye, and the wording, as well as the column heading ("EDM Element Descriptions") will inform the user that this is the target element definition.	Yes. Once the user recognizes the target element definition for what it is, they will realize they should read it through, ensuring that the source element metadata would appropriately fit within the definition and explanation of the target element.	Yes. As the user compares the source element informaiton and definition to that of the target element, they will begin to verify the semantic similarities expected based upon the syntactic similarities of the element names.
Compares to givenName element. Determines a single dc:creator element should include both the givenName and the lastName.	Yes. After making another choice for the target element to be mapped to the source element, the user will want to discover whether the definition, allowable values, etc. align with those of the source element.	Yes. Once the dc:creator element is selected, the definition appears next to the dropdown menu. The change of information on the screen will draw the user's eye, and the wording, as well as the column heading ("EDM Element Descriptions") will inform the user that this is the target element definition.	Yes. Once the definition appears, the user will again want to compare it to the definition and information they have available for the source element.	Yes. As the user compares the source element information and definition to that of the target element, they see that dc:creator is a match for givenName and lastName, which combine to provide the author.
Returns to lastName row.	Yes. When the user realizes that givenName and lastName should be combined to form dc:creator, they will want to specify this is both for givenName and lastName.	Yes. The user knows that the source elements are listed in rowsin the table, and know the left most comlumn provides the element names.	No. The user may be unsure of how they are expected to specify that givenName and lastName are supposed to be in the same instance of the dc:creator field. They may expect there to be a different way to specify this combination staying the the givenName row.	No. The user may be unsure that this is the correct path, and therefore unsure whether returning to the lastName row is progress.
Clicks the dropdown menu.	Yes. The user is making a mapping between two metadata schema, and will know that they must make element mappings. After looking over the information for a source element, they will next want to look at the potential target elements available to find the best match.	Yes. The dropdown menu is direclty to the right of the element definition, and as the user continues to follow the left-to-right eye motion, they will see the dropdown menu for the row and along with its default value ("Select an EDM element") and realize this is how they proceed with the mapping. The column heading will also give an indication that this is where the EDM element is chosen.	Yes. The user will associate this row's dropdown menu with mods:title. The dropdown menu is right next to the element definition and is a logical next step. A dropdown menu in HTML is associated with selecting a choice, making a decision; the user will see this is where their input comes in.	Yes. When the user clicks on the dropdown box, all of the EDM elements will appear in the menu to select from.
Selects dc:creator.	Yes. The user has already decided that givenName and lastName are supposed to be in dc:creator, and will therefore want to make this choice after clicking the dropdown menu.	Yes. The dc:creator option is listed in the dropdown menu. After the user clicks into the dropdown menu, they will again read down the options until they find the dc:creator element.	Yes. The user will associate selecting dc:creator in the dropdown menu with choosing dc:creator as the target element for lastName.	Yes. The definition of dc:creator will appear next to the dropdown menu, verifying that is was selected as the target element for last:name.
In the notes field for both givenName and lastName rows, the user writes: "Create single dc:creator field with contents of givenName and lastName, separated by a space."	Yes. The user will want to specify that givenName and lastName elements should combine into a single dc:creator element to make an appropriate mapping.	Yes. The user will see the Notes text field directly adjacent to the taget element dropdown menu and definition at the end of the row.	No. The user will be very confused as to how to specify that givenName and lastName elements should be combined into a single field. Even if they decide the Notes Field is the correct action to take, they may not know whether to put a note in both source element rows, or just one -- and which one at that.	No. If the user decides that putting a descriptive note in the text fields of both of the elements, they may feel that they have successfully specified the mapping. However, there also may be some uncertainty as to whether this is the appropriate and complete way to specify this mapping of two source elements into a single field.

Many-to-One Choice				
First, the user identifies the row for the location element.	Yes. Given that the user is attempting to make a mapping between their source metadata and EDM, they will anticipate making element-by-element mappings, and will therefore focus on one element at a time. This example has them focusing on the location element.	Yes. The righthand columns all start out identical in each row, the two columns on the left are unique in each row. A user used to using tables will look to the top row for column headings and read columns left to right. Looking at the left most column, the user will be able to identify that each row represents a separate element, and be able to locate the row for the location element	Yes. The user will see the element name location, and realize that this is the information concerning the source metadata and where they should start when attempting to map to the target schema.	Yes. When the user identifies the mods:title element row, they will realize realize they are on the right path, and are now able to move toward the next step of the mapping
The user reads through the metadata information. The user reads the definition of the source element.	Yes. After identifying the correct row for the element they are trying to map, the user may want to know more precise information about the use of the element field, and would likely want to consult the element definition prior to making a mapping decision.	Yes. The element description is directly to the right of the element name making it very visible for the user. It follows a logical left-to-right eye movement.	Yes. Again, the definition for the source element appears directly to the right of the element name and instance information. Given that all of the elements are in a single, vertical column, the user will view the contents of this row as pertinent to the mods:title source element.	Yes. As the user is aware of the purpose of mapping the source schema to the target schema, they will understand that the definition of the element can be an important aid tot eh mapping. Reading through the official definition, they may be able gain a greater understanding of any nuances of the element use.
Clicks on dropdown menu.	Yes. The user is making a mapping between two metadata schema, and will know that they must make element mappings. After looking over the information for a source element, they will next want to look at the potential target elements available to find the best match.	Yes. The dropdown menu is direclty to the right of the element definition, and as the user continues to follow the left-to-right eye motion, they will see the dropdown menu for the row and along with its default value ("Select an EDM element") and realize this is how they proceed with the mapping. The column heading will also give an indication that this is where the EDM element is chosen.	Yes. The user will associate this row's dropdown menu with mods:title. The dropdown menu is right next to the element definition and is a logical next step. A dropdown menu in HTML is associated with selecting a choice, making a decision; the user will see this is where their input comes in.	Yes. When the user clicks on the dropdown box, all of the EDM elements will appear in the menu to select from.
Locates and selects edm:currentLocation in the dropdown menu.	Yes. When presented with the options for target elements to map to, the user will know they must select a potential mapping, and will want to look through the names for a syntactically or semantically similar element name that may match with the source location element.	Yes. When the menu is clicked, all of the possible EDM choices dropdown. The user will understand they must select one.	Yes. Seeing edm:currentLocation, the user would likely select it, assuming there would be a significant similarity between it and the source element location. However, if there is any uncertainty, the user may not initially know that the tool will display the definition of the selected element. While many of the element names provide some guidance as to what values they will contain, detailed definitions are really required to make an informed decision on element mappings. They may not realize that the definition will appear as an aid to making the desicion and believe that they must make the mapping based on the target element names alone. However, if this is the case, they would still likely select edm:currentLocation for its syntactic similarity to mods:title.	Yes. After edm:currentLocation is selected, the element definition will appear to the right of the dropdown menu. Showing that their selection was recognized.
***Element definition appears.* User reads information and sees that edm:currentLocation is not repeatable.**	Yes. After making a preliminary choice for the target element to be mapped to -- edm:currentLocation, the user will want to ensure that the definition, allowable values, etc. align with those of the source element.	Yes. Once the edm:currentLocation element is selected, the definition appears next to the dropdown menu. The change of information on the screen will draw the user's eye, and the wording, as well as the column heading ("EDM Element Descriptions") will inform the user that this is the target element definition. The information about the repeatability of the element is at the beginning of the definition in bold	Yes. Once the user recognizes the target element definition for what it is, they will realize they should read it through, ensuring that the source element metadata would appropriately fit within the definition and explanation of the target element.	Yes. As the user compares the source element informaiton and definition to that of the target element, they will begin to verify the semantic similarities expected based upon the syntactic similarities of the element names.
Looks back at source metadata information. Sees that there are more location elements than there are records.	No. Though the user will likely see that edm:currentLocation is not repeatable, they may not register that they should (and are able) to see at least to some extent in the mapping tool whether the source element location is unique in each record, and may therefore not think to check this.	No. Even if the user does think to check whether location is unique in each source record, they may not realize the mapping tool provides some help with this. The number of records is separated from the number of occurrences of the elements. In addition, it is possible that there are multiple records that have no location element, while others have multiple occurrences -- and therefore the number of occurrences of the location element could still be lower than the number of records, thus providing no guidance as to whether the element is repeatable. The user would possibly be able to look at the sample values and see the relationship between them	No. The tool does not make it obvious how one can check whether an element is repeated within a single record in the source metadata. The user will likely have difficulty ascertaining this even if they realize that they must.	Yes. If the user is able to identify from the source metadata information that the location element is repeatable in individual records, then they will identify that there is an issue that needs to be resolved -- the lack of repeatability of the target element edm:currentLocation. While this also adds complication to their task of specification, they will also see the identification of this issue as a step towards creating a quality mapping.

Looks at sample values. Sees that building information is in first occurrence, coordinates in second occurrence.	Yes. When the user realizes that only one of the multiple occurrences of the location element can be mapped to edm:currentLocation, they will want to figure out whether there is a good way to choose between the multiple occurrences.	No. There is a lot of information in the first source element cell. They may or may not be aware of the sample values present.	No. The user may not see the sample values as a potential way to figure out how to decide which occurrence of location should be mapped to edm:currentLocation, or, more specifically, what the differences are between the multiple occurrences are.	Yes. If the user is able to locate and use the sample values to determine the difference between the two occurrences of the source element location in individual records, they will see that one is coordinates and one is the building. Armed with this information, they will then know they need simply to decide which of these is the most appropriate choice for edm:currentLocation.
Refers to edm:currentLocation definition which emphasizes precise location over building information. Decides second occurrence should be used in edm:currentLocation. Other occurrence could be put in a dc:description to minimize data loss, or simply left out of the mapping.	No. Armed with the information concerning the different occurrences of the source element location, the user will want to decide which better correlates with the edm:currentLocation target element and should therefore be mapped to it. The user will want to choose between the two, and probably decide that the more precise option should be used in edm:currentLocation. However, exactly how the user will decide to handle the extra location field is uncertain, as they may be unsure of what to do with this information.	Yes. They have already seen and read through the definition for edm:currentLocation once, and will be able to locate it again easily within the same row they have been working in. The user will see that they must make a choice between the two options based on the edm:currentLocation definition.	Yes. They know they must look at more information about edm:currentLocation, and they know where the definition is, and will return to the definition to consider its relationship to the different occurrences of location.	Yes. As the user reads through the definition of edm:currentLocation, they will make a decision between the two kinds of locations present in the source element location. Once they make their choice, they will see they have made progress towards completing a quality mapping. Once they make the choice, they will see that they are closer to completing the mapping.
Writes in Note field: "Second occurrence of location element [coordinates] mapped to edm:currentLocation."	Yes. After making a choice of which location occurrence will appear in the nonrepeatable edm:currentLocation element, they will want to specify this choice within the mapping.	Yes. The user will see the Notes text field directly adjacent to the target element dropdown menu and definition at the end of the row.	No. The user may be confused as to how to specify which occurrence of the source location element should be used in the edm:currentLocation element. They may not realize that using the Notes Field is the correct action to take.	Yes. If the user realizes that this should be specified in the notes field and types in the note specifying which occurrence of the location element should be mapped to edm:currentLocation, they will see that the specification is complete.

Unmatched Source Elements				
Locate source element row.	Yes. Given that the user is attempting to make a mapping between their source metadata and EDM, they will anticipate making element-by-element mappings, and will therefore focus on one element at a time. This example has them focusing on the mods:title element.	Yes. The righthand columns all start out identical in each row, the two columns on the left are unique in each row. A user used to using tables will look to the top row for column headings and read columns left to right. Looking at the left most column, the user will be able to identify that each row represents a separate element, and be able to locate the row for the mods:title element	Yes. The user will see the element name info, and realize that this is the information concerning the source metadata and where they should start when attempting to map to the target schema.	Yes. When the user identifies the mods:title element row, they will realize realize they are on the right path, and are now able to move toward the next step of the mapping
Read through metadata information and definition.	Yes. After identifying the correct row for the element they are trying to map, the user may want to know more precise information about the use of the element field, and would likely want to consult the element definition prior to making a mapping decision.	Yes. The element description is directly to the right of the element name making it very visible for the user. It follows a logical left-to-right eye movement.	Yes. Again, the definition for the source element appears directly to the right of the element name and instance information. Given that all of the elements are in a single, vertical column, the user will view the contents of this row as pertinent to the mods:title source element.	Yes. As the user is aware of the purpose of mapping the source schema to the target schema, they will understand that the definition of the element can be an important aid tot eh mapping. Reading through the official definition, they may be able gain a greater understanding of any nuances of the element use.
Click dropdown menu.	Yes. The user is making a mapping between two metadata schema, and will know that they must make element mappings. After looking over the information for a source element, they will next want to look at the potential target elements available to find the best match.	Yes. The dropdown menu is direclty to the right of the element definition, and as the user continues to follow the left-to-right eye motion, they will see the dropdown menu for the row and along with its default value ("Select an EDM element") and realize this is how they proceed with the mapping. The column heading will also give an indication that this is where the EDM element is chosen.	Yes. The user will associate this row's dropdown menu with mods:title. The dropdown menu is right next to the element definition and is a logical next step. A dropdown menu in HTML is associated with selecting a choice, making a decision; the user will see this is where their input comes in.	Yes. When the user clicks on the dropdown box, all of the EDM elements will appear in the menu to select from.
Look through EDM elements and their definitions.	Yes. The user will be looking for a target element that can be mapped to the source element. They will want to examine their options for an appropriate match -- both the names and the definitions of the target elements.	No. After clicking into the dropdown menu, the user will infer that they are meant to make a choice of one of the options below. They may not realize that the definitions will appear to assist them.	No. There are many EDM elements to choose from, and if none are syntactically or semantically similar to the source element in name, the user may flounder on how to proceed. The process of eliminating all possible mappings by looking at the definitions of each individually could be very time consuming, and potentially frustrating, for the user.	No. As the user looks through the various target element options, they may not feel that they are making any progress towards task completion, except that they are eliminating all the inappropriate mapping choices one at a time.
Decide there is no appropriate exact or approximate match. Select No match in the dropdown menu.	Yes. If there is absolutely no viable target element to map to the source element, the user will conclude that the mapping just isn't possible (though if they are attempting to make a relative mapping, they may of course never come to this conclusion and merely choose a target element that has any relation to the source element).	No. After looking through the options on the dropdown menu, it is very likely that they would have noticed the No Match option is available.	Yes. If the user is aware that the No Match option is in the dropdown menu, once they decide there is no viable match for the souce element, they will know that this is the option they should choose.	Yes. The user will have made a decision on the mapping of the source element, even if it is to have no mapping to the target schema, and the values will then be lost.

Overlap				
Locate the row for the info source element.	Yes. Given that the user is attempting to make a mapping between their source metadata and EDM, they will anticipate making element-by-element mappings, and will therefore focus on one element at a time. This example has them focusing on the mods:title element.	Yes. The righthand columns all start out identical in each row, the two columns on the left are unique in each row. A user used to using tables will look to the top row for column headings and read columns left to right. Looking at the left most column, the user will be able to identify that each row represents a separate element, and be able to locate the row for the mods:title element	Yes. The user will see the element name info, and realize that this is the information concerning the source metadata and where they should start when attempting to map to the target schema.	Yes. When the user identifies the mods:title element row, they will realize realize they are on the right path, and are now able to move toward the next step of the mapping
Read the metadata information and element description.	Yes. After identifying the correct row for the element they are trying to map, the user may want to know more precise information about the use of the element field, and would likely want to consult the element definition prior to making a mapping decision.	Yes. The element description is directly to the right of the element name making it very visible for the user. It follows a logical left-to-right eye movement.	Yes. Again, the definition for the source element appears directly to the right of the element name and instance information. Given that all of the elements are in a single, vertical column, the user will view the contents of this row as pertinent to the mods:title source element.	Yes. As the user is aware of the purpose of mapping the source schema to the target schema, they will understand that the definition of the element can be an important aid tot eh mapping. Reading through the official definition, they may be able gain a greater understanding of any nuances of the element use.
Click the dropdown menu.	Yes. The user is making a mapping between two metadata schema, and will know that they must make element mappings. After looking over the information for a source element, they will next want to look at the potential target elements available to find the best match.	Yes. The dropdown menu is direclty to the right of the element definition, and as the user continues to follow the left-to-right eye motion, they will see the dropdown menu for the row and along with its default value ("Select an EDM element") and realize this is how they proceed with the mapping. The column heading will also give an indication that this is where the EDM element is chosen.	Yes. The user will associate this row's dropdown menu with mods:title. The dropdown menu is right next to the element definition and is a logical next step. A dropdown menu in HTML is associated with selecting a choice, making a decision; the user will see this is where their input comes in.	Yes. When the user clicks on the dropdown box, all of the EDM elements will appear in the menu to select from.
Examine chioces and decide on dc:description.	Yes. The user clicked on the dropdown menu to expand it to see the potential choices for mapping from the info source element. They will want to locate a target element that is appropriate to be mapped to the info element, and will likely settle on dc:description at some point.	Yes. After clicking into the dropdown menu, the user will infer that they are meant to make a choice of one of the options below. Locating the appropriate choice may take a few tries, but info and description have enough of a semantic association that the user will likely attempt dc:description at some point.	No. It might take the user a while to settle on dc:description as a potential mapping for their source element info. The user may not initially know that the tool will display the definition of the selected element. While many of the element names provide some guidance as to what values they will contain, detailed definitions are really required to make an informed decision on element mappings. They may not realize that the definition will appear as an aid to making the desicion and believe that they must make the mapping based on the target element names alone. However, after making any choice at all, the user will see that the definition of the element appears, and will know that they will receive more guidance. The user will likely eventually give dc:description a try.	Yes. After dc:description is selected, the element definition will appear to the right of the dropdown menu. Showing that their selection was recognized.
***Definition appears.* Compare definition to source definition and info. Realize the element match some of the time.**	Yes. After making a preliminary choice for the target element to be mapped to the info source, the user will want to ensure that the definition, allowable values, etc. align with those of the source element.	Yes. Once the dc:description element is selected, the definition appears next to the dropdown menu. The change of information on the screen will draw the user's eye, and the wording, as well as the column heading ("EDM Element Descriptions") will inform the user that this is the target element definition.	Yes. Once the user recognizes the target element definition for what it is, they will realize they should read it through, ensuring that the source element metadata would appropriately fit within the definition and explanation of the target element.	Yes. As the user compares the source element informaiton and definition to that of the target element, they will begin to verify the semantic similarities expected based upon the syntactic similarities of the element names.
Decide on a relative mapping, typing notes in the Notes field to extrapolate the divergences and justification of decision.	No. The user will want to make a decision on how to handle the overlap situation between the source and target elements (see Additional Notes section to the left for the other possible options the user must consider). The user may not realize that they are meant to explain the overlap issues by writing an explicit note in the Note field. They may simply make the decision on the mapping and move on.	Yes. The Notes field is highly visible next to the dropdown menu. The user is very likely to see it and know that they are able to include notes in the field.	Yes. If the user decides to explain the mapping decision and issues with the overlap situation fully, they will see the Notes field next to the dropdown menu and dc:description definition as the optimal way to include these thoughts in the mapping specification. The presence of the Notes field in itself could possibly prompt the user to add in such notes.	Yes. Once the user has made their decision on the mapping, and written any explanations and issues in the notes section, they will be at the end of the row and done with the mapping of this source element. And can therefore move on to the next element.

Content Requirement - data type/value range				
Locate the row for the date source element.	Yes. Given that the user is attempting to make a mapping between their source metadata and EDM, they will anticipate making element-by-element mappings, and will therefore focus on one element at a time. This example has them focusing on the mods:title element.	Yes. The righthand columns all start out identical in each row, the two columns on the left are unique in each row. A user used to using tables will look to the top row for column headings and read columns left to right. Looking at the left most column, the user will be able to identify that each row represents a separate element, and be able to locate the row for the mods:title element	Yes. The user will see the element name info, and realize that this is the information concerning the source metadata and where they should start when attempting to map to the target schema.	Yes. When the user identifies the mods:title element row, they will realize realize they are on the right path, and are now able to move toward the next step of the mapping
Read the metadata inforamtion and element description.	Yes. After identifying the correct row for the element they are trying to map, the user may want to know more precise information about the use of the element field, and would likely want to consult the element definition prior to making a mapping decision.	Yes. The element description is directly to the right of the element name making it very visible for the user. It follows a logical left-to-right eye movement.	Yes. Again, the definition for the source element appears directly to the right of the element name and instance information. Given that all of the elements are in a single, vertical column, the user will view the contents of this row as pertinent to the date source element.	Yes. As the user is aware of the purpose of mapping the source schema to the target schema, they will understand that the definition of the element can be an important aid tot eh mapping. Reading through the official definition, they may be able gain a greater understanding of any nuances of the element use.
Click the dropdown menu.	Yes. The user is making a mapping between two metadata schema, and will know that they must make element mappings. After looking over the information for a source element, they will next want to look at the potential target elements available to find the best match.	Yes. The dropdown menu is direclty to the right of the element definition, and as the user continues to follow the left-to-right eye motion, they will see the dropdown menu for the row and along with its default value ("Select an EDM element") and realize this is how they proceed with the mapping. The column heading will also give an indication that this is where the EDM element is chosen.	Yes. The user will associate this row's dropdown menu with mods:title. The dropdown menu is right next to the element definition and is a logical next step. A dropdown menu in HTML is associated with selecting a choice, making a decision; the user will see this is where their input comes in.	Yes. When the user clicks on the dropdown box, all of the EDM elements will appear in the menu to select from.
Examine choices and decide on dc:date.	Yes. For an element like date, which is relatively straight forward, the user is likely to scan through the available target elements to find one that looks like it would contain a date.	Yes. After clicking into the dropdown menu, the user will infer that they are meant to make a choice of one of the options below. Looking down the list, the user will see dc:date as an option.	Yes. Seeing dc:date, the user would select it, assuming there would be a significant similarity between it and the date source element. However, if there is any uncertainty, the user may not initially know that the tool will display the definition of the selected element. While many of the element names provide some guidance as to what values they will contain, detailed definitions are really required to make an informed decision on element mappings. They may not realize that the definition will appear as an aid to making the desicion and believe that they must make the mapping based on the target element names alone. However, if this is the case, they would still likely select dc:date for its syntactic similarity to date.	Yes. After dc:date is selected, the element definition will appear to the right of the dropdown menu. Showing that their selection was recognized.
Definition appears. Compare definition to source definition and metadata information.	Yes. After making a preliminary choice for the target element to be mapped to the date source element, the user will want to ensure that the definition, allowable values, etc. align with those of the source element.	Yes. Once the dc:date element is selected, the definition appears next to the dropdown menu. The change of information on the screen will draw the user's eye, and the wording, as well as the column heading ("EDM Element Descriptions") will inform the user that this is the target element definition.	Yes. Once the user recognizes the target element definition for what it is, they will realize they should read it through, ensuring that the source element metadata would appropriately fit within the definition and explanation of the target element.	Yes. As the user compares the source element informaiton and definition to that of the target element, they will begin to verify the semantic similarities expected based upon the syntactic similarities of the element names.
Realize that sample values do not conform to the recommended format for dc:date values.	Yes. The dc:date definition provides the recommended date format. They will want to check to see if the recommendation matches with the formating of the source metadata.	No. The format inforamtion for the dc:date element is in the element definition next to the dropdown menu. The sample values for the source metadata are listed underneath the date element name, which are visible but may not have been noticed with the plethora of information present.	Yes. If the user sees that the sample values are present in the first cell, after having read the formating recommendation, they will want to compare the values to the format.	Yes. The user will see that the source values do not conform to the format recommendations for dc:date, which tells them they must either og against the recommendation or create a mapping specification that specifies that these values must be modified.
Write in Note field the recommended format for date values.	Yes. The user will want to specify that the source metadata values must be modified to adhere to the recommended format for dc:date.	Yes. The Notes field is highly visible next to the dropdown menu. The user is very likely to see it and know that they are able to include notes in the field.	No. The user may not realize that the way they are meant to specify that the values in the source element should be reformatted is by writing an explicit note in the Note field. They may expect something more concrete -- something that is machine readable, for instance. If they do realize this is where they are meant to specify such instructions, they will likely type in an adequately specific note to inform the mapping specification.	No. If the user realizes that this should be specified in the Notes field and types in the note specifying the reformatting of the source element values to the recommended format of dc:date, they may see that the specification is complete. However, they may also be uncertain as to whether this is the the correct way to complete the specification.

Conversion Combinations				
Locate the source element day.	Yes. Given that the user is attempting to make a mapping between their source metadata and EDM, they will anticipate making element-by-element mappings, and will therefore focus on one element at a time. This example has them focusing on the mods:title element.	Yes. The righthand columns all start out identical in each row, the two columns on the left are unique in each row. A user used to using tables will look to the top row for column headings and read columns left to right. Looking at the left most column, the user will be able to identify that each row represents a separate element, and be able to locate the row for the mods:title element	Yes. The user will see the element name info, and realize that this is the information concerning the source metadata and where they should start when attempting to map to the target schema.	Yes. When the user identifies the mods:title element row, they will realize realize they are on the right path, and are now able to move toward the next step of the mapping
Read the metadata information and element description. Realize this is one element of a three-part composit to create a full date.	Yes. After identifying the correct row for the element they are trying to map, the user may want to know more precise information about the use of the element field, and would likely want to consult the element definition prior to making a mapping decision.	Yes. The element description is directly to the right of the element name making it very visible for the user. It follows a logical left-to-right eye movement. The information of the element would include information about the three-part date, and the sample values of the element show that only the day of the date is included in the values.	Yes. Again, the definition for the source element appears directly to the right of the element name and instance information. Given that all of the elements are in a single, vertical column, the user will view the contents of this row as pertinent to the day source element.	Yes. As the user is aware of the purpose of mapping the source schema to the target schema, they will understand that the definition of the element can be an important aid tot eh mapping. Reading through the official definition, they may be able gain a greater understanding of any nuances of the element use.
Click dropdown menu.	Yes. The user is making a mapping between two metadata schema, and will know that they must make element mappings. After looking over the information for a source element, they will next want to look at the potential target elements available to find the best match.	Yes. The dropdown menu is direclty to the right of the element definition, and as the user continues to follow the left-to-right eye motion, they will see the dropdown menu for the row and along with its default value ("Select an EDM element") and realize this is how they proceed with the mapping. The column heading will also give an indication that this is where the EDM element is chosen.	Yes. The user will associate this row's dropdown menu with mods:title. The dropdown menu is right next to the element definition and is a logical next step. A dropdown menu in HTML is associated with selecting a choice, making a decision; the user will see this is where their input comes in.	Yes. When the user clicks on the dropdown box, all of the EDM elements will appear in the menu to select from.
Examine choices and decide on dc:date.	Yes. For an element like date, which is relatively straight forward, the user is likely to scan through the available target elements to find one that looks like it would contain a date.	Yes. After clicking into the dropdown menu, the user will infer that they are meant to make a choice of one of the options below. Looking down the list, the user will see dc:date as an option.	Yes. Seeing dc:date, the user would select it, assuming there would be a significant similarity between it and day element, which is a part of a date. However, if there is any uncertainty, the user may not initially know that the tool will display the definition of the selected element. While many of the element names provide some guidance as to what values they will contain, detailed definitions are really required to make an informed decision on element mappings. They may not realize that the definition will appear as an aid to making the desicion and believe that they must make the mapping based on the target element names alone. However, if this is the case, they would still likely select dc:date, as it is very relatable semantically to day.	Yes. After dc:date is selected, the element definition will appear to the right of the dropdown menu. Showing that their selection was recognized.
***Definition appears.* Compare definition to source definition and metadata information.**	Yes. After making a preliminary choice for the target element to be mapped to the date source element, the user will want to ensure that the definition, allowable values, etc. align with those of the source element.	Yes. Once the dc:date element is selected, the definition appears next to the dropdown menu. The change of information on the screen will draw the user's eye, and the wording, as well as the column heading ("EDM Element Descriptions") will inform the user that this is the target element definition.	Yes. Once the user recognizes the target element definition for what it is, they will realize they should read it through, ensuring that the source element metadata would appropriately fit within the definition and explanation of the target element.	Yes. As the user compares the source element information and definition to that of the target element, they will begin to verify the semantic similarities expected based upon the syntactic similarities of the element names.
Determine day, month, and year source elements must be combined into recommended dc:date format. Write this specification in the Note field.	Yes. After seeing the definition and sample values for the day element, the user will be aware that this is only one portion of the full date, and must be combined with otehr elements to create the dc:date. The dc:date definition also states the recommended format for its values, and the user will realize they should adhere to this standard.	Yes. The Notes field is highly visible next to the dropdown menu. The user is very likely to see it and know that they are able to include notes in the field.	No. The user will know that they must combined the day, month and year elements in the recommended format to complete the dc:date field, but may be unaware of how they should specify this. The user may not realize that the way they are meant to specify this is by writing an explicity note in the Note field. They may expect something more concrete -- something that is machine readable, for instance. If they do realize this is where they are meant to specify such instructions, they will likely type in an adequately specific note to inform the mapping.	No. If the user decides that putting a descriptive note in the text fields of both of the elements, they may feel that they have successfully specified the mapping. However, there also may be some uncertainty as to whether this is the appropriate and complete way to specify this mapping of three source elements into a single field.
Locate month and year element rows. For each, choose dc:date and copy the Note field.	No. The user may not realize they likely should complete the same information for the other two source elements used in this mapping.	Yes. The user will be able to locate the two source elements the same way the day element was located. They will be fimiliar enough with the layout, having already interacted with the row for the day element.	No. Again, the user could be very uncertain as to whether to complete the information in the rows for the month and year source elements, as well as for the day source element.	No. The user still may be unsure as to whether the specification is complete if done this way.