

Developing Metadata Application Profiles

Mariana Curado Malta

*Polytechnic of Oporto, Portugal & Algoritmi Center, University
of Minho, Portugal*

Ana Alice Baptista

Algoritmi Center, University of Minho, Portugal

Paul Walk

University of Edinburgh, UK

A volume in the Advances
in Web Technologies and
Engineering (AWTE) Book Series



www.igi-global.com

Published in the United States of America by

IGI Global

Information Science Reference (an imprint of IGI Global)

701 E. Chocolate Avenue

Hershey PA 17033

Tel: 717-533-8845

Fax: 717-533-8661

E-mail: cust@igi-global.com

Web site: <http://www.igi-global.com>

Copyright © 2017 by IGI Global. All rights reserved. No part of this publication may be reproduced, stored or distributed in any form or by any means, electronic or mechanical, including photocopying, without written permission from the publisher.

Product or company names used in this set are for identification purposes only. Inclusion of the names of the products or companies does not indicate a claim of ownership by IGI Global of the trademark or registered trademark.

Library of Congress Cataloging-in-Publication Data

Names: Malta, Mariana Curado, 1969- editor. | Baptista, Ana Alice, editor. | Walk, Paul, 1968- editor.

Title: Developing metadata application profiles / Mariana Curado Malta, Ana Alice Baptista, and Paul Walk, editors.

Description: Hershey, PA : Information Science Reference, [2017] | Includes bibliographical references and index.

Identifiers: LCCN 2016056939 | ISBN 9781522522218 (hardcover) | ISBN 9781522522225 (ebook)

Subjects: LCSH: Metadata--Standards.

Classification: LCC Z666.7 .D48 2017 | DDC 025.3--dc23 LC record available at <https://lcn.loc.gov/2016056939>

This book is published in the IGI Global book series Advances in Web Technologies and Engineering (AWTE) (ISSN: 2328-2762; eISSN: 2328-2754)

British Cataloguing in Publication Data

A Cataloguing in Publication record for this book is available from the British Library.

All work contributed to this book is new, previously-unpublished material. The views expressed in this book are those of the authors, but not necessarily of the publisher.

Chapter 7

Using Reverse Engineering to Define a Domain Model: The Case of the Development of a Metadata Application Profile for European Poetry

Mariana Curado Malta

Polytechnic of Oporto, Portugal & LINHD-UNED, Spain

Paloma Centenera

LINHD-UNED, Spain

Elena Gonzalez-Blanco

LINHD-UNED, Spain

ABSTRACT

This chapter presents the early stages of a metadata application profile (MAP) development that uses a process of reverse engineering. The context of this development is the European poetry, more specifically the poetry metrics and all dimensions that exist around this context. This community of practice has a certain number of digital repertoires that store this information and that are not interoperable. This chapter presents some steps of the definition of the MAP Domain Model. It shows how the developers having as starting point these repertoires, and by means of a reverse engineering process are modeling the functional requirements of each repertoire using the use-case modeling technique and are analyzing every database

DOI: 10.4018/978-1-5225-2221-8.ch007

Using Reverse Engineering to Define a Domain Model

logical models to extract the conceptual model of each repertoire. The final goal is to develop a common conceptual model in order to use it as basis, together with other sources of information, for the definition of the Domain Model.

INTRODUCTION

Comparative literary studies have always been a source of new discoveries which enlighten the perspectives of other related disciplines, such as history, archaeology or sociology. It is sometimes difficult, however, to get results in the philological field, as the sources to compare are uneven, follow different historical, linguistic and literary traditions and do not have many elements in common to take them as a reference or starting point.

Poetry studies have suffered from this reality, as each different cultural tradition has followed an independent way, where no standards were adopted for terminology or classification. Each literary school has modelled a different system that looked to be the most suitable for its own problems. Communication between different languages and literatures has been almost scarce, even from the critical point of view. The result is a fragmentary puzzle which includes different traditions, languages, literary and poetic schools not possible to analyze using the same methods and straightforward paths to compare poetic forms.

From the point of view of literary analysis, the studies on metrics and poetry were first linked to grammar and rhetoric, and were not considered independent as “ars metrica” or “ars poetica” until the 14th century (Gómez Redondo, 2001). During the Middle Ages and the Renaissance, the powerful influence of Latin as the language of culture made scholars inherit the terminology of Classical treaties and apply it to Romance languages, regardless of their different way of structuring. When vernacular theories start to arise, each particular school makes up its own terminology and classification system. This multiplicity leads to paradoxical and complex situations, such as the creation of conceptual genres that only exist in some literatures.

A special case to illustrate this problem is the phenomenon of metrical repertoires, catalogues which aimed at gathering all poetic and metrical features in any of the different literary tradition. They show the way in which researchers measure and classify poems, counting syllables, accents, rhythm and rhymes to define the essential elements of the poem structure, its musicality and the type of contents that it shapes. A digital poetry metrics repertoire is a tool that gives account of metrical and rhythmical schemes of either a poetical tradition or school gathering a long corpus of poems, which are defined and classified by their main characteristics.

The first poetic repertoires appeared as books at the end of the 19th century with the works of Gaston Raynaud (1884), Gotthold Naetebus (1891), and Pillet and Carstens (1933), followed by some other important books after the Second World War, as the classic work of Frank (1953-1957) on Provençal troubadours' poetry, and with the editions of printed metrical repertoires in Old French lyrics Mölk and Wolfzettel (1972), in Italian Antonelli (1984), Solimena (2000), Zenari (1999), Pagnotta (1995), and Gorni (2008), in the Hispanic philology Tavani (1967), Parra i Blasco (1992), and Gómez Bravo (1999), in the German, Toubert (1975) and Brunner et al. (1986-2007).

The evolution of technologies in the last decades had a strong impact in all disciplines. Humanities were also transformed by these changes after Padre Busa in 1949 published the first digital database: the *Corpus Thomisticum* supported by IBM technology. After that date, many projects –especially related with philology and linguistic studies- tried to combine the technological innovations with their data building digital databases and resources which would make life easier for researchers and future users. Although they did not consider themselves “digital humanists,” those researchers were the pioneers of the creation of a new research area.

Technological advances made possible to create a new generation of repertoires, in which time of research is considerably reduced and the user experience much better than in the previous paper books. The first digital poetic repertoire online was the *RPHA*: Répertoire de la Poésie hongroise ancienne jusqu'à 1600¹. Galician researchers created MedDB: Base de datos da Lírica profana galego-portuguesa²; Italian researchers digitalized BEdT: Bibliografia Elettronica dei Trovatori³; the Nouveau Naetebus for French narrative poetry⁴, the Oxford Cantigas de Santa María Database⁵ the Analecta Hymnica Digitalia⁶, the Dutch Song Database⁷, the Corpus Rhythmorum Musicum⁸, the Repertorio della tradizione poetica italiana dai Siciliani a Petrarca⁹, the Digital Edition of the Index of Middle English Verse¹⁰, The Last Song of the Troubadours¹¹, the Czech Versification Research Group¹², and finally “Repertório Métrico Digital de la poesia medieval castellana” (ReMetCa)¹³ started in 2011 by the POSTDATA research group, which is the last piece of this poetic multilingual Babel-like puzzle.

The latest mentioned projects were created as web-based resources and databases and interoperability among poetic repertoires is not simple, as there are not only technical issues involved, but also conceptual and terminological problems: each repertoire belongs to its own poetical tradition and each tradition has developed an idiosyncratic analytical terminology in a different and independent way for years. The result of this uncoordinated evolution is a bunch of varied terminologies to explain analogous metrical phenomena through the different poetic systems, whose correspondences have been hardly studied.

Using Reverse Engineering to Define a Domain Model

The approaches to digital poetic repertoires have been always linked to a philological purpose, following traditional research methodologies which have been considered a “close reading” approach to texts and corpora (Moretti 2013), even though the advanced in digital repertoires have been great in the last decades. The POSTDATA project is presented in this chapter as a Digital Humanities frame in which “distant reading” is the approach to compare hundreds of results at the same time, the research questions and the ways of representing knowledge becomes different¹⁴. This Digital Humanities approach is what POSTDATA team aims at achieving by publishing the data of the digital repertoires as linked open data (LOD).

To achieve this goal of publishing the data of the digital repertoires as LOD there is the need to have a common model for the data. POSTDATA is now developing a metadata application profile for the European Poetry, and this chapter presents the process and techniques used to define a common domain model. This chapter proceeds as follows: Section 2 presents the motivation for the development of a metadata application profile for the European Poetry, Section 3 shows how developers used a reverse engineering process 1) to model Functional Requirements and; 2) to analyze existent databases to build conceptual models as a process to develop the Domain Model. Section 4 presents recommendations and Section 5 concludes and presents future work.

Motivation

The POSTDATA project focus in the interoperability issue between the digital poetry metrics repertoires in Europe. These digital repertoires are mainly in the Web of Documents¹⁵, but there are also some stored locally. The repertoires were developed in a specific technology along time, and store the data in their own silos. This data is at the moment locked in the silos of information of each repertoire, not available freely to be compared and to be used by intelligent machines that could infer over data. In fact the interoperability problem exists because the technological solutions used for building each digital repertoire and data model is very different from each other.

POSTDATA is a project that aims to provide the means to open this data stored in many databases of the Web of Documents publishing it as Linked Open Data (LOD). All the data trapped in the silos of information of each digital repertoire needs to be mapped to a common semantic model. This semantic model is in fact a metadata application profile (MAP), a construct that enhances interoperability (Nilsson, Baker, & Johnston, 2009)pp.

POSTDATA is now, at the time of writing this chapter, developing a MAP for the European Poetry. Next chapter will show in detail how the first steps of this development are being implemented.

Developing Functional Requirements and Domain Model

Introduction

As mentioned previously the context of POSTDATA is the digital repertoires of poetry. This kind of repertoires may sometimes contain the text of the poem and information related to authors, manuscripts and editions all of them related to the poems. They also have other dimensions that are related to the poems depending on the mission of the repertoire and context. They have distinct corpus of poems and information that are, as example:

- Relations between poems and songs, and metrics of the songs,
- Relations between poems and miracles that are described in poems,
- Relations between poems and places that are referred in poems,
- Relations between poems and people, real persons that existed and other that are imaginary like for example, Don Quijote de la Mancha.

The development of a MAP is a crucial task for a community of practice. This development should be structured, and integrate, since the early phases of development, elements of all representative members of the community. Commonly the organizations differ in organization-type, location, culture and in the language they speak. To find a common ground of understanding in such an environment becomes a huge challenge. Adding to this, a MAP development is often done in complex settings that are very open, in contrast with the development of software that serves a certain organization that is protected inside its walls of context, culture and language, where requirements can be elicited more easily using very well known techniques. In a MAP development, designers will never know in fact the total reach of the MAP, the community of practice that the MAP serves can be very well defined but there will be always a degree of uncertainty - to elicit requirements is not easy in such uncertainty.

The POSTDATA team thinks that the existence of a method for the development of a MAP may help to address all the referred challenges. Recent studies say that there is no method for the development of MAPs (see Curado Malta & Baptista (2013c)), in order to address this issue Curado Malta and Baptista (2013a, 2013b) have been working on the definition of a method for the development of metadata application profiles (Me4MAP). The POSTDATA team is using Me4MAP16 to develop the referred MAP.

MeMAP defines the need to develop a set of deliverables in order to define a MAP of quality. These deliverables are the output of a set of activities that are globally organized by the Singapore Stages. This name comes after the seminal

Using Reverse Engineering to Define a Domain Model

document “Singapore Framework for the development of dublin core application profiles - see Nilsson, Baker, and Johnston (2008)”. S1, the Singapore Stage 1, develops the functional requirements of the MAP and S2, the Singapore Stage 2, defines the Domain Model. Every stage includes a certain number of activities to achieve the main deliverable of every Singapore Stage. Other Stages exist but it is not the aim of this chapter to explain in detail Me4MAP rather than to address the way POSTDATA defined the Functional Requirements and the Domain Model, main deliverables of S1 and S2, respectively.

POSTDATA knows the existence of twenty-two digital repertoires, twenty-one are in the Web of Documents and one is local. The owners of the repertoires were contacted and documentation from seventeen repertoires was received. The following sources of information were used to develop the Functional Requirements and the Domain Model:

- Study of the functionalities of the Web interfaces of the twenty-one repertoires;
- Study of the seventeen data models:
 - Eleven of the repertoires implemented the database in a relational database management system
 - Four of the repertoires are implemented in a XML database system
 - One of the repertoires uses Perl objects to store data
 - The local repertoire is implemented in a stylesheet
- Survey to final users of the repertoires to know what kind of things they would like to do in an interface (e.g. Webpage, App, SPARQL endpoint) that deals with information about poems
- Two case studies of digital repertoires that are being built at the same time the MAP is being developed. These case studies also provide information on how to model the data that needs to be captured

Next sections will present in detail the techniques used to develop the Functional Requirements and the Domain Model deliverables.

Modeling Functional Requirements

A MAP development process is guided by data since data is the essential part of a MAP. The main objective to define functional requirements is to identify the data and relationships between the data that support the identified functionalities during the functionalities analysis. This section explains how the analysis of the Websites of the repertoires was implemented in order to model the functional requirements of the MAP.

Using Reverse Engineering to Define a Domain Model

In the case of the MAP for the European Poetry, the functional requirements definition was developed using a reverse engineering process (Müller et al., 2000) (Pressman, 2015). Chikofsky and Cross wrote in the 90's about reverse engineering and design recovery taxonomies. They defined reverse engineering as “the process of analysing a subject system to identify the system's components, their interrelationships, and create representations of the system in another form or at a higher level of abstraction” (Chikofsky and Cross, 1990, pp.15). This process reconstructs abstract models from implementation, the abstract models help to understand the different aspects of the system to be analysed.

The reverse engineering process was very important in the past for the maintenance of legacy systems (Bianchi, Caivano, Marengo & Visaggio, 2003) (Hassan, Qamar, Hassan & Waqas, 2015) (Sneed, 1995). Nowadays this process is very useful in domain engineering processes that aim to define the commonality and the variability of the software product line (Linden Schmid and Rommes, 2007) (Pohl, Blöckle & van Der Linden, 2005). Software product lines reduce the cost of customized applications, building them by assembling reused components. These components are previously built in a domain engineering process. The aim of domain engineering processes is that the final products are highly reusable. This aim is aligned with our objective because a MAP serves a wide range of users, all very different. A MAP is in fact highly reused. This is the reason why the authors decided to use reverse engineering in the process of developing the Domain Model.

This reverse engineering process has three steps:

Step 1: Identification of the functionalities that exist in the Websites of the repertoires

Step 2: Use-case model development

Step 3: Identification and description of the data elements of the use-case model

In Step 1 developers navigated through the Webpages of the repertoires and analyse the functionalities these pages have. The Oxford Advanced Learner's Dictionary¹⁷ defines functionality as “the range of functions that a computer or other electronic system can perform.” In fact, a functionality is the set of functions executed by the system that transform input data in output data in a way that the user obtains a useful result. The aim of Step 1 is the rebuilding of the model of the requirements of the Website. When the Webpages are simple, this work can be done with no tools, just analyzing the Webpages and taking notes in a structured text document. In more complex Websites the analysis can be made resorting to tools that help to automatize the process (Bouillon, Vanderdonckt & Chow, 2004).

In Step 2 the use-case modeling technique is used to express the functional requirements (Jacobson, Booch, Rumbaugh, & Booch, 1999). This is also the technique suggested by Me4MAP. A use case is “a unit of functionality expressed as a

Using Reverse Engineering to Define a Domain Model

transaction among actors and the subject” (Rumbaugh, Jacobson, & Booch, 2004). Actors are entities outside the system that interact with it. A use-case model “characterizes the kinds of behavior provided by an entity, such as a system, subsystem, or class, in its interactions with outside entities. Outside entities are actors of the entity” (Rumbaugh et al., 2004). The use-case model is expressed in a UML use case diagram. Such a diagram is built by doing the description of the actor interaction with the system in the different use-case scenarios. This description allows more or less formal methods of representation and formulations (Rumbaugh et al., 2004) of UML interactions. Satechart diagrams, sequence diagrams, collaboration diagrams, or informal text descriptions can be used. In the present case, the POSTDATA team has used informal text descriptions (Somé, 2009) because the functionalities identified in the Websites are simple, and with this representation the use case model is easy to understand.

Table 1 presents a template for the description of use-cases. Each use-case is identified by a number and a name. Then the actor that performs the Use-Case is identified and a short description of the use-case function is defined. The main flow is set by enumerating the interactions between the actor and the system in the main scenario. The main scenario is the most used, alternative flows can exist at any moment of the flow, they should be described in the second column (Alternative flow) and only if a specific action is very different compared to the main flow. There may exist more than one alternative flow. All alternative flows should be numbered. Each interaction should have a number, and the alternative flow of a specific action should have the same number one level up (e.g See Table 1: Main flow - 1, Alternative flow(1) - 1.1.; Main flow - 2, Main flow - 3).

In Step 3 data elements identified in the navigation of the repertoire’s Web-pages are documented (see the activity defined in Step 1). Table 2 shows a template of the documentation, it consists in a description of the characteristics of the data elements such as:

Table 1. Template for a textual description of a Use-Case

Number: The number assigned to the Use Case Use Case Name: The name of the Use Case Actor: The main actor of the Use Case. It is the actor who launches the Use Case Description: Short description of the Use Case function	
Main flow	Alternative flow (1)
1. Description of interactions between user and system in the main scenario 2. 3.	1.1 Description of the interaction in a alternative scenario. Only if a specific step is very different compared to the main flow

Table 2. Template of the data elements documentation

Window number/name: Number or name of the window that is being described. Data elements			
Label	Cardinality	Searchable	Link
The label of the data element in the window	e.g 1; 0 - *; 1 - *	Yes/No	Yes/No

- **Cardinality:** If the data element is or not repeatable, is or not optional
- **Searchable:** If the data element is in a search filter
- **Linkable:** If the data element is or not a link to navigate to another window

The content of Table 2 helps to understand the functionalities of the Webpages. This information will feed activity S2 – developing the Domain Model, since it defines data elements to integrate this model.

Each repertoire case has its own model of functional requirements. To finalise S1 – Developing the Functional Requirements the POSTDATA team has to analyse all models and define a common model for the Functional Requirements of the MAP. This model integrates the requirements of all repertoires and new requirements that come from the survey and the case studies already mentioned. Also, since the MAP is going to be used by LOD technologies which are different from the ones of the Web of Documents, all the requirements should be submitted to a reengineering process to be adapted to this new reality and to take advantage of the potential of the available technology. This adaption is under study in order to understand the implications of the new reality.

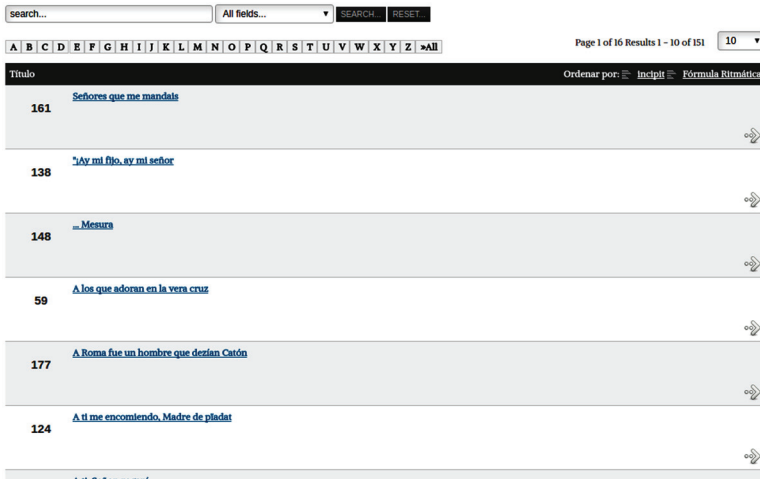
Next paragraphs will present a real case by using the ReMetCa¹⁸ repertoire as example to show this reverse engineering process.

Step 1: Functionalities Identification

A user accesses the Website of the ReMetCA to look for metrical information about a certain poem. To be able to access the repertoire the user clicks in the “repertoire” menu option on the top menu. The Webpage presents all poems of the repertoire (see Figure 1) by showing a list of title poems. By default they are ordered by title (also called “incipit”). The user has the possibility to search a poem, or group of poems, by “Fórmula Rimática,” by “Title” or by “all fields” (it is not clear the meaning of “all fields” since later in the process there are other fields displayed in the result of a search and it is not clear if those fields were object of the search too). The Webpage also has a functionality that allows to filter the list of titles of poems

Using Reverse Engineering to Define a Domain Model

Figure 1. Front page of the ReMetCa repertoire



by letter. For this functionality, the Webpage presents a list of letters (all letters of the alphabet) where the user can click on the required letter to activate the filter.

In order for the user to do a search there is the need of doing three actions (see Figure 2):

1. Insert a string in the search text box
2. Choose a field
3. Press the search button

The result of the search is a list of poem titles that correspond to the criteria of the search. This information is displayed in the same window where the search is performed. If the user clicks in a title of a poem of the list, the system displays all the metrical information about that specific poem (see result in Figure 3). If the user clicks the print button (see (1) in Figure 3) the system prints the metrical information of the selected poem (see Figure 4).

Step 2: Requirements Modeling

There are three main use-cases in ReMetCa (see Figure 5).

The use-cases' description is displayed in Table 3, 4 and 5.

Each use-case presents a function of the system. The Functional Requirements are explicit in those use-case descriptions. Step 3 shows what data elements are needed to support these functionalities.

Using Reverse Engineering to Define a Domain Model

Figure 2. Window 1: REMETCA front page with the actions identified by numbers (1), (2) and (3)

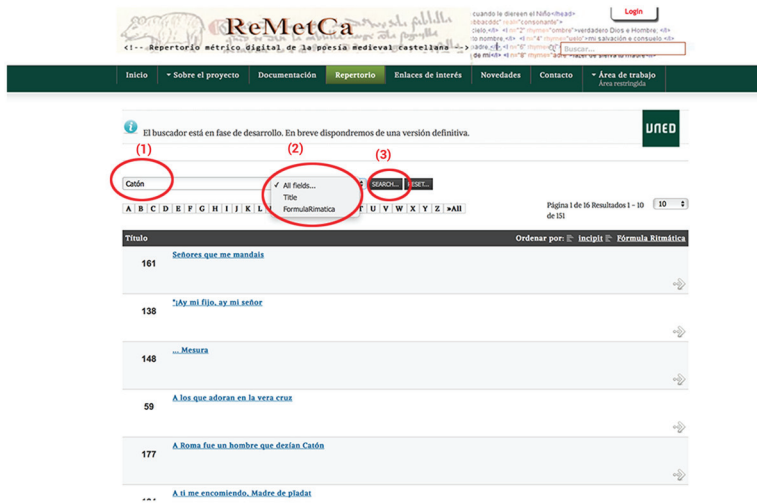
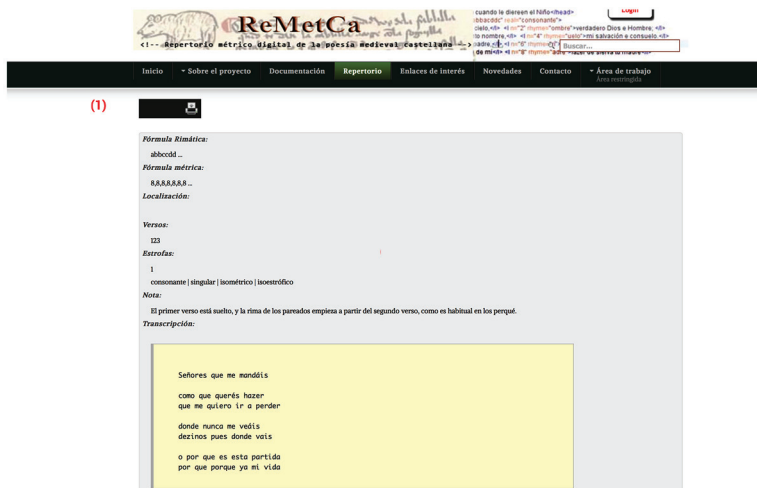


Figure 3. Window 2: Detailed metrical information about a poem



Step 3: Describing Data Elements


Each use-case explicits a certain number of data elements, the task here is to extract that information using the template of Table 2. The description of the data elements identified in the use-cases of Step 2 is presented in Tables 6, 7 and 8.

Using Reverse Engineering to Define a Domain Model

Figure 4. Window 3: Detailed metrical information about a poem to be printed

Señores que me mandais

8,8,8,8,8,8 ...



Localizacion:

Item: 949

Verses: 123

Stanza: 1

FormulaRimatica: abbccdd ...

Note: El primer verso está suelto, y la rima de los pareados empieza a partir del segundo verso, como es habitual en los perquè.

Figure 5. UML use-case diagram for ReMetCa

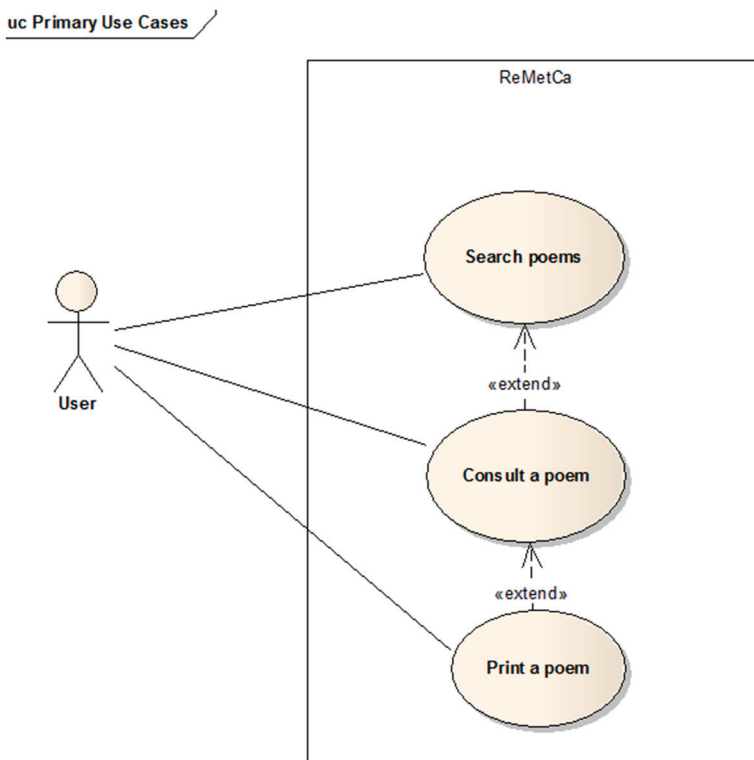


Table 3. Description of ReMetCa Use-case 001 (Search poems)

Number: 001 Use Case Name: Search poems Actor: User Description: Allows the user to find a list of titles of poems			
Main flow	Alternative flow (1)	Alternative flow (2)	Alternative flow (3)
1. The user selects the option “Repertorio” in the top menu			
2. The system retrieves and shows all “incipit” of the poems sorted by “incipit” (See Figure 1)			
3. The user writes a string in the search box, selects a field, and presses the “Search” button - Window 1 (See Figure 2)	3.1 The user selects to order the list by “incipit” or “Fórmula Rimática”	3.2 The user presses a letter	3.3. The user presses a “incipit” in the list
4. The system retrieves and shows the list of “incipit” of the poems that match the search criteria (sorted by “incipit” - default)	4.1 The system retrieves and shows the list of “incipit” of the poems ordered by the selected criteria: “incipit” or “Fórmula Rimática”	4.2 The system retrieves and shows the list of “incipit” of the poems which start by the selected letter	4.3. Extension to the use-case 2 - Consult a poem
5. Return to step 3			

Table 4. Description of ReMetCa Use-case 002 (Consult a poem)

Number: 002 Use Case Name: Consult a poem Actor: User Description: Allows the user to consult metrical information of a poem	
Main flow	Alternative flow
1. The user clicks in one of the poem’s “incipit” of the list of Use-Case 001 – Search poems	
2. The system shows all metrical information about the poem - Window 2 (see Figure 3)	
3. The user presses the print button. (optional - Extension to the use-case 3 – Print a poem).	

Using Reverse Engineering to Define a Domain Model

Table 5. Description of ReMetCa Use-case 003 (Print a poem)

Number: 003 Use Case Name: Print a poem Actor: User Description: Allows the user to print the metrical information of a poem	
Main flow	Alternative flow
1. The user clicks in the print button	
2. The system retrieves the information and opens a new window – Window 3 (see Figure 4) with the information	

Table 6 presents the data elements of Window 1 (Figure 1). In that window only one data element is identified: “Incipit.” This data element is not repeatable, that is, one poem only has one “incipit”; it is and is obligatory; and it is linkable, meaning that the user can click that element to navigate to Window 2, (which is the function “Consult a poem”).

Table 7 presents the data elements of Window 2 (see Figure 3). In this Window the system presents, after the user clicks in the title of a poem, metrical information of that single poem. Seven data elements can be clearly identified, none of them is

Table 6. Description of the data elements of Window 1

Window number/name: Window 1. Data elements			
Label	Cardinality	Searchable	Link
Incipit	1	Yes	Yes

Table 7. Description of the data elements of Window 2

Window number/name: Window 2. Data elements			
Label	Cardinality	Searchable	Link
Fórmula Rimática	1	No	No
Fórmula Métrica	1	No	No
Localización	1	No	No
Versos	1	No	No
Estrofas	1	No	No
Notas	0 - 1	No	No
Transcripción	1	No	No

repeated (that is, each data element only contains one single piece of information per poem). Only “Notas” is optional, all other elements are obligatory. No data element is neither searchable neither linkable in Window 2 since in this Window the user cannot search anything and cannot click in any of those elements to navigate to a new page.

Table 8 presents the data elements of Window 3 (see Figure 4). In this Window the system presents, after the user clicks the print button (in Window 2), printable metrical information of one single poem that was consulted in Window 2. Eight data elements can be clearly identified, none of them is repeated. Only “Notas” is optional, all other elements are obligatory. No data element is neither searchable neither linkable in this Window.

With the conclusion of Step 3 the Functional Requirements of ReMetCa are modeled and the data elements that support those functionalities are identified. According to Me4MAP this information will feed next activity S2 – developing the Domain Model.

In the POSTDATA case, the work-team is resorting also to other information to feed S2: a survey to final users and two case studies that are developing repertoires at the same time POSTDATA is developing the MAP. It is not in the aim of this chapter to explain how these sources of information were used to feed S2.

Developing the Domain Model

According to Me4MAP the Domain Model development is fed by the Functional Requirements, the deliverable developed in S1 activity. But Me4MAP also adds that this development may also resort to other types of information the community

Table 8. Description of the data elements of Window 3

Window number/name: Window 3.			
Data elements			
Label	Cardinality	Searchable	Link
Incipit	1	No	No
Fórmula Métrica	0 - 1	No	No
Localización	0 - 1	No	No
Item	0 - 1	No	No
Versos	0 - 1	No	No
Estrofas	0- 1	No	No
Fórmula Rimática	0 - 1	No	No
Notas	1	No	No

Using Reverse Engineering to Define a Domain Model

of practice might have, like documentation or any other type of information. In the case of the POSTDATA project the community of practice has twenty repertoires in the Web of Documents and a local repertoire in a Worksheet. This section explains how the logical models of the repertoires were analyzed and how, using a reverse engineering process, they were converted in conceptual models.

A domain model is a conceptual model that is used to explicit the concepts that exist in a certain universe of discourse. A Domain Model is used to explicit the kind of concepts the universe has and, through the use of properties, how these concepts are defined. A MAP is a semantic model that will be based in a conceptual model, assigning 1) terms of RDF vocabularies to the concepts and properties and 2) adding constraints to the terms.

This section uses real examples to show how the conceptual models were created having as starting point:

1. A MySQL dump file
2. A XML xsd file
3. A perl file with a sample of objects
4. A stylesheet with sample data

The first example presented here is the repertoire “Repertório Métrico Digital de la poesia medieval castellana” (ReMetCa), the same repertoire used to explain how the reverse engineering process was implemented (see previous sub-section). ReMetCa has a relational database implemented in MySQL. The database responsible sent to POSTDATA a MySQL dump with the database schema. It was possible to recreate the database locally and extract the logical relational model (see Figure 6) using the software MySQL Workbench¹⁹.

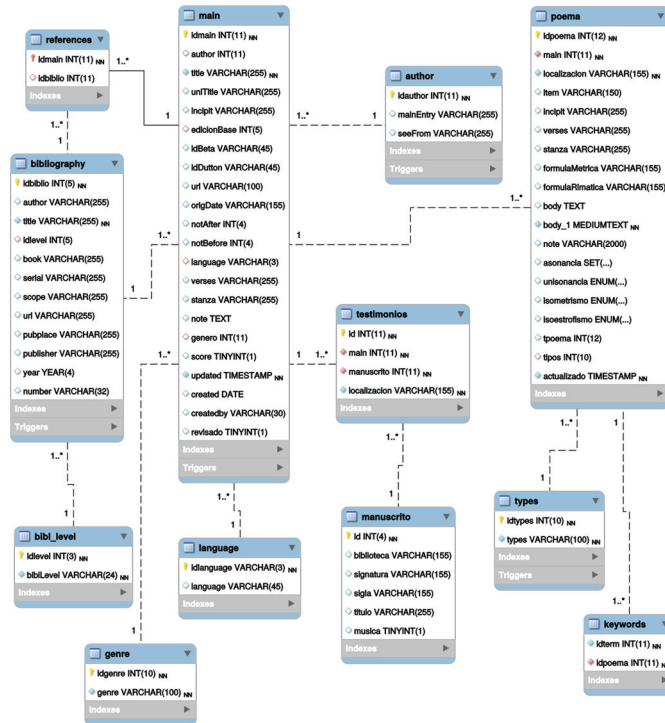
The process starts by the analysis of the logical model in order to understand the concepts that each table represent.

Follows a list of the POSTDATA team reflexions and assumptions:

- A type of thing is called “table” in a relational logical model, “concept” in a conceptual model, and “class” in a semantic model
- In the process of development of a conceptual model from a logical model there is the need to eliminate details that exist because of the implementation:
 - Primary keys are not part of the “world of concepts” that conceptual models model. It is a fact that in a semantic model classes have identifiers (URIs) to implement the need to identify univocally a resource, but in a conceptual model that is not necessary
 - Foreign keys exist to explicit the relations between tables. A conceptual model explicit these relations through a line connecting concepts that

Using Reverse Engineering to Define a Domain Model

Figure 6. Logical relational model of the repertoire “Repertorio Métrico Digital de la poesía medieval castellana”



are related, and a number that defines the cardinality of the relation. A semantic model explicits the relations between classes using object properties, the line and cardinality in the concept model is sufficient to define later the object properties. Foreign keys do not exist, then, in the conceptual models

- Tables whose properties are only keys, only exist in order to implement many-to-many relations. These tables are details of the representation and do not appear in conceptual models
- Value tables (tables with lists of possible values for a property) are not concepts, they are instead properties of another concept with multiple or single cardinality

These last two items are introduced in a relational logical model in the process of normalization. Since this is a reverse engineer process where the goal is to arrive to a conceptual model in the context of a future semantic modeling, these items are deleted from the conceptual model.

Using Reverse Engineering to Define a Domain Model

Analyzing the logical model, the POSTDATA team identified:

- Four main tables: “main” (represents a whole poem), “poema” (represents a strophic pattern in a poem), “author” (represents the author of a poem), “bibliography” (represents a book that refers to the poem) and “manuscrito” (represents the manuscript where poems were written). These tables will be represented in the conceptual diagram as four different concepts, and the concepts will have the same properties the tables of the logical model have.
- Two tables created to express the many-to-many relationships:
 - “References” relates “main” to “bibliography”: it expresses that one poem can be referred by many books and a book can refer many poems. This table only has one key so it does not exist in the conceptual model;
 - “Testimonios” relates “main” to “manuscrito”: it expresses that one poem can be present in more than one manuscript and that a manuscript can have many poems. This table besides the primary key (composed by three properties) has also a property that defines the location of the poem in the manuscript. This table is represented in the conceptual model only with one property (“localizacion”).
- Five tables that represent lists, these tables might be a way to control the introduction of terms by the user. In fact, all tables have two properties: a primary key and another property with the same name of the name of the table (or very similar). These tables are “genre,” “bibl_level,” “language,” “keyword” and “types.” These tables don’t appear as concepts in the conceptual model since they can be replaced by a property (see Table 9). “genre,” “bibl_level,” “language” and “keyword” can be repeatable properties, “type” not. In the MAP definition all these fields will be terms with a vocabulary encoding scheme as range.

Table 9. Mapping of the tables that represent controlled terms to properties in the REMETCA Conceptual Model

Table & Property Logical Model	Concept & property Conceptual Model	Property Cardinality
language - language	main - language	0 - *
genre - genre	main - genre	0 - *
bibl_level - bibl_level	main - bibl-level	0 - *
type - type	StanzaPattern - type	0 - 1
keywords - idterm	StanzaPattern - keyword	0 - *

Using Reverse Engineering to Define a Domain Model

The final goal of this work is to obtain a Domain Model that will represent the community of practice universe of discourse. To achieve that, it is important to standardize the names these community gives to things since the beginning of the process. The same concept should have the same name. So, when analyzing the logical models, the POSTDATA team started immediately defining common names among all conceptual models for all concepts and properties that represent the same thing. This is the reason why when comparing the logical and the respective conceptual model names are different (neither tables, neither properties). This process is iterative, after building one model, the work-team goes back to the previous ones already developed, revise names and decide if the names are appropriate or need to change according to the new findings in the new models developed. This is done during the analysis of all models.

Table 10 presents the correspondence between Tables and Concepts in the process of building the conceptual model of ReMetCa. A similar table exists for each table that maps the properties names of the tables to the properties names of the concepts. All MAP development project should document very well all the process (as Me4MAP states). These mappings will help the owners of the repertoires to understand the correspondences between the original logical model and the final semantic model of the MAP, and will help developers in the future to migrate the data from one model to another.

The POSTDATA team used stylesheets to document the process. Figure 7 presents an excerpt of ReMetCa documentation. The stylesheet has a first row with the name of the database, a second row with the name of the first table to be documented and the correspondent name of the concept in English. If the table is not mapped the field is left empty. The stylesheet has three more columns:

Table 10. Correspondence between the tables of the Logical Model and the concepts of the Conceptual Model of ReMetCa

Logical Model Table	Conceptual Model concept
main	Work
poem	StanzaPattern
author	Author
testimonio	Manifestation
manuscrito	Manuscript
bibliography	RefBibliography

Using Reverse Engineering to Define a Domain Model

Figure 7. An excerpt of the stylesheet of ReMetCa Documentation

	A	B	C	D
1	Name of Database	REMETCA		
2				
3	Entity Logical Model	author	Entity Conceptual Model	Author
4	Property Logical Model	Property Conceptual Model	Description	
5	ldauthor	PRIMARY KEY		
6	mainEntry	name	Citado según el Índice de Autoridades de la Biblioteca Nacional, aunque quitando la fecha que aparece después	
7	seeFrom	identifier		
8				
9	Entity Logical Model	main	Entity Conceptual Model	Work
10	Property Logical Model	Property Conceptual Model	Description	
11	idmain	PRIMARY KEY		
12	author	FOREIGN KEY REF AUTHOR		
13	title	title	Campo obligatorio. Seguimos el índice de Autoridades de la Biblioteca Nacional, en su pestaña "título"	
14	uriTitle	uriTitle	Campo no obligatorio. Recoger otros títulos o variantes por los que pueda ser conocido el poema. Dejar en blanco si coincide con el anterior.	
15	incipit	incipit		
16	genero	genre	Referencia Hsmetca. Seguimos el índice de la monografía Historia de la Métrica Medieval Castellana. Escoger el grupo al que pertenece la obra.	See Work.genre
17	edicionBase	FOREIGN KEY REF REFBIBLIOGRAPHY	Edición que sirve de base al poema sobre el cual se realiza el análisis métrico. Remite a la tabla BIBLIOGRAFIA	
18	idBeta	identifierBeta	Número de 4 dígitos: Ej: 8932 del índice de Philobiblon: http://bancroft.berkeley.edu/philobiblon/ son los cuatro dígitos que van después de la referencia BETA	
19	idDutton	identifierDutton	Número de 4 dígitos: Ej: 2897. Solamente para los poemas de cancionero recogidos en el índice de Dutton. Para el resto, dejar en blanco. Puede comprarse la referencia en http://cancionerovirtual.liv.ac.uk/pinchingdo en "access the editions"	
20	uri	uriText	Recoger en el caso de que el texto completo (o parcial) esté disponible en web, aunque presente variantes con respecto a la edición que manejamos. Trasar de recoger siempre que sea posible la uri del texto completo disponible en web, aunque no sea una edición muy fiable.	
21	origDate	origDate		
22	notAfter	notAfter	Fecha ante quem. 4 dígitos numéricos. Horquilla aproximada, si no se sabe con exactitud, recoger la horquilla más amplia	
23	notBefore	notBefore	4 dígitos numéricos	
24	language	language	Seleccionar castellano, a no ser que sea un texto que tenga gran parte escrita en gallego. En cualquier caso, si la lengua predominante es el castellano, las variantes o interpolaciones simplemente se citan en el campo nota.	

Column 1: The names of the properties, as they are called originally in the database;

Column 2: The names of the properties of the concepts. New names in English are given with the concern of standardization. If a property is not mapped in the conceptual model the correspondent cell is left empty. This can happen in cases where there is a bad modulation. In case the property moves to another concept, that should be noted in the correspondent cell. Keys are noted too.

Column 3: A description of the property

There is a fourth column in case the property has a range with a set of controlled terms (e.g. the property “genre” of the concept “Work”). In the ReMetCa Case these terms are taken from the database documentation available online. In other cases the controlled terms can be found in the MySQL dump file or in the combo-boxes of the public Web interfaces of the repertoires.

To document the set of controlled terms a new sheet is created with the list of terms, the name of the sheet is given according to the following convention: “NameOfConcept.NameOfControlledVocabulary” (see Figure 8).

The conceptual model of ReMeTca is presented in Figure 9.

The data elements identified in the activity S1 – developing the Functional Requirements (see Tables 6, 7 and 8) are all present in this model. There are other concepts and properties that do not appear in Tables 6, 7 and 8. In fact, S1 did not

Figure 8. A controlled set of terms documented

	A	B	C
1	Principios poéticos		
2	Épica y juglaresca		
3	Debate		
4	Noticieros e historiográficos		
5	Poesía clerical		
6	Poesía aljamiada y clerecía rabínica		
7	Poesía lírica castellana tradicional		
8	Poesía lírica castellana cortés		
9	Poesía hagiográfica y didáctica		
10	Poesía cortesana y cancioneril		
11	Oraciones y textos litúrgicos		
12	Poemas historiográficos		
13	Prosimetra: exégesis, alegoría y ficción sentimental		
14	Romancero		
15	Poesía italianizante		
16	Dramaturgia		
17			
18			
19			
20			
21			
22			

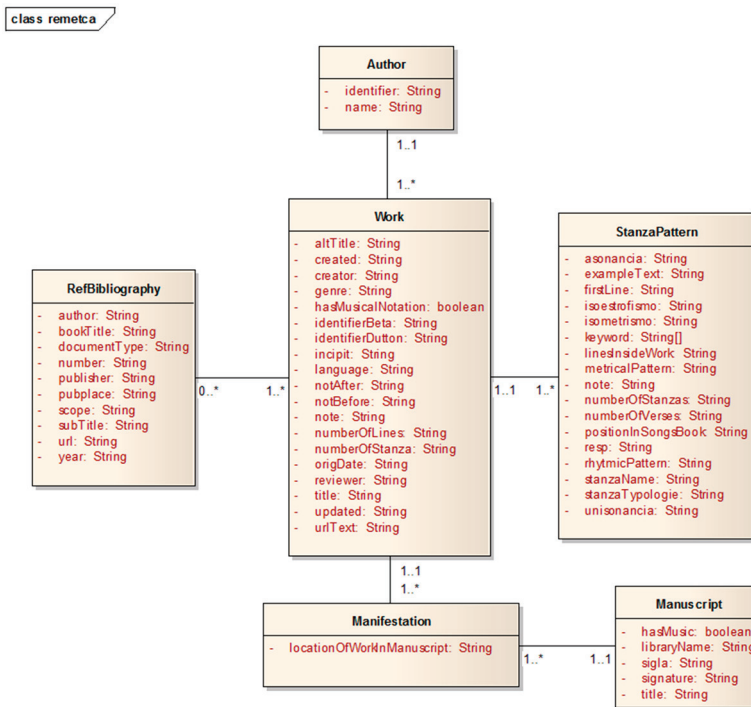
bring any new information to feed this model. The database has more information than the one used in the public Website. Since the Website is in construction, POSTDATA expects other functionalities to be added in the future to the ones identified (use-case 1, 2 and 3), these new functionalities might use the concepts and properties of the Conceptual Model presented in Figure 9. Some of the properties of the conceptual model might be used also to implement functionalities of the Back-end Website.

The second example presented here is the repertoire “Digital Edition of the index of Middle English Verse” (DIMEV)²⁰. This repertoire uses a XML database to store data. The repertoire’s responsible sent to POSTDATA a xsd file with the XML schema of the model. The software Oxygen XML editor²¹ was used to create a visual image (see Figure 15 in Appendix) of the scheme. This XML scheme uses a subset of TEI²² as data model to represent texts in a digital form, in particular the module that TEI has for poetry²³.

The process starts by analyzing the XML schema in order to understand the concepts that each element represent. We identified four concepts: “Work,” “Witness,”

Using Reverse Engineering to Define a Domain Model

Figure 9. Conceptual model of the repertoire “Repertório Métrico Digital de la poesía medieval castellana”



“Source” and “Interval.” The main concept is the description of a “Work” (poem), a “Work” can have many “Witnesses” and a “Witness” can have many “Sources,” a “Source” can have many “Intervals.” “Work” has the elements “alpha,” “descNote,” “nimev,” “imev,” “name,” “identifier,” used only once per “Work,” and “author,” “ghost,” “title,” “subject,” “verseForm,” “versePattern,” that are repeatable. “Witness” has the elements “allLines,” “firstLines,” “identifier,” “ilust,” “lastLines,” “msAuthor,” msTitle,” “music” and “sourceName” that are used only once per “Witness,” and “facsimile” and “edition” that are repeatable. The “Source” concept has two elements “key” and “prefix” not repeatable and the “Interval” concept has two elements “start” and “end” also not repeatable.

The UML class diagram of the conceptual model of the repertoire “DIMEV” is presented in Figure 10.

The third example presented here is the repertoire “Versologie”²⁴. This repertoire stores data in perl objects. The responsible for this database sent to POSTDATA a script file in perl with a subset of data that represents the model (see Figure 11 for an excerpt of the perl file).

Using Reverse Engineering to Define a Domain Model

Figure 10. Conceptual MODEL of the repertoire “Digital Edition of the index of Middle English Verse”

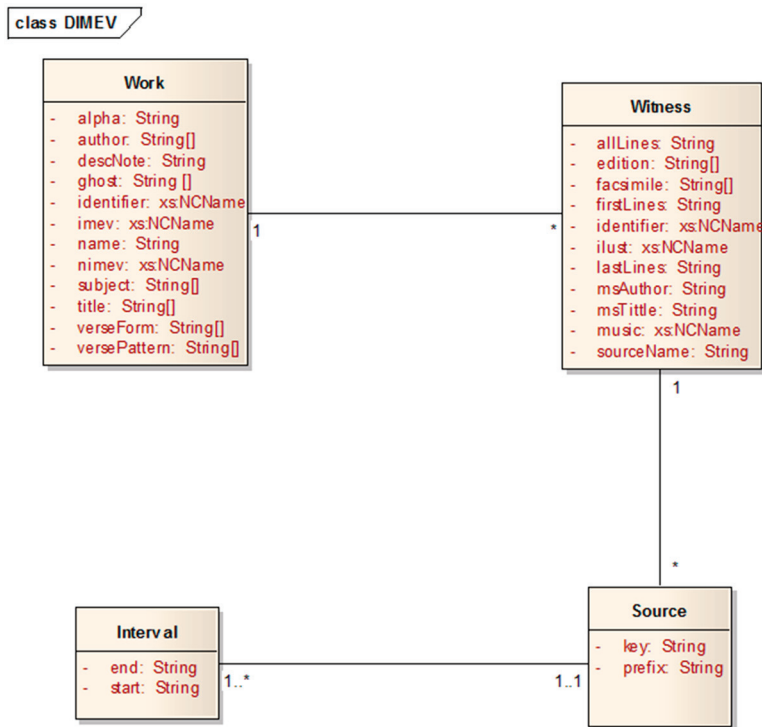


Figure 11. An excerpt of a Versologie perl script where it can be seen the data stored as objects

```

'head' => {
# ===== IDS =====
'id' => {
# Id of a book from which the poem comes from
'book' => '0001',
# Id of a poem (unique in entire corpus - first four numbers correspond to the Id of a book
'poem' => '0001-0000-0000-0001-0000',
},
# ===== GENERAL BIBLIOGRAPHIC METADATA =====
'biblio' => {
# Title of a book from which the poem comes from
'title' => 'Na zemi a na nebi',
# Subtitle of a book
'subtitle' => 'Básně',
# Year of publication of a book
'year' => 1900,
# Place where the book was published
'place' => 'Praha',
# Publisher of a book
'publisher' => 'Unie',
# First, second, third... edition of a book
'edition' => 1,
# Name of the library from which the book have been digitized & the signature of a book in library
'signature' => 'Národní knihovna ČR, Praha: 54 H 2287'
}

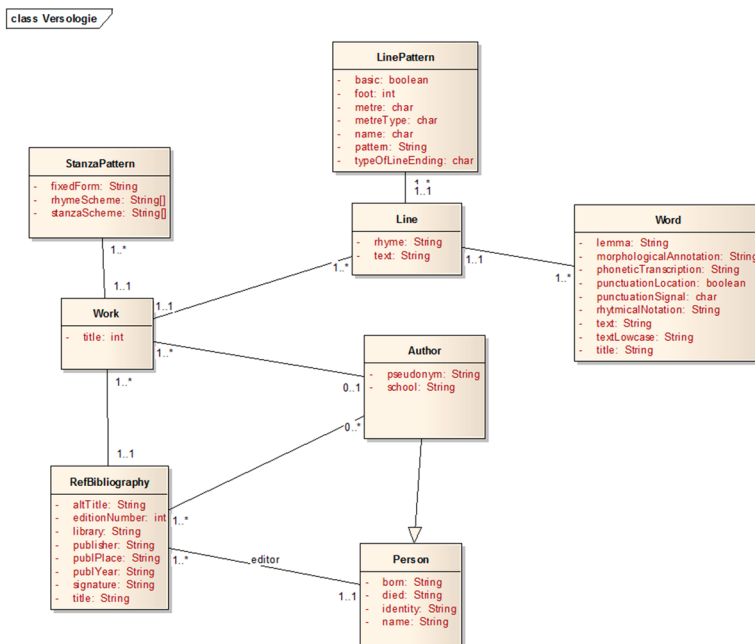
```

Using Reverse Engineering to Define a Domain Model

The UML Class diagram of the conceptual model of “Versologie” is presented in Figure 12. The file presented in Figure 11 shows a top concept that is “Work” (poem), this top concept has several other concepts that are related to “Work”: “RefBibliography,” “Author,” “StanzaPattern” and “Line.” “Line” is related to “LinePattern” and “Word.” “Person” is a class used to define a person that can be an author (sub-class of “Person” defined as the concept “Author”) or an editor of books that refer “Work,” these books that refer “Work” are defined in the diagram as the concept “RefBibliography”. A book also can have an editor that is expressed through the relationship between “Person” and “RefBibliography.” “StanzaPattern” is a pattern that exists inside a “Work.” The concept “Line” represents every line of the poem (“Work”). Each “Line” has a specific pattern (“LinePattern”) and is composed by words (“Word”).

Both repertoires “DIMEV” and “Versologie” are on the Web of Documents so the POSTDATA team has also done the analysis of the Websites using the same reverse engineering process reported in the example of ReMetCa to model the functional requirements of those repertoires. There is no space in this chapter to present the two processes since the process is exactly the same and would not bring anything new. Both analyses did not bring any new element to the conceptual models defined in S2 – developing the Domain Model activity.

Figure 12. Conceptual model of the repertoire “Versologie”



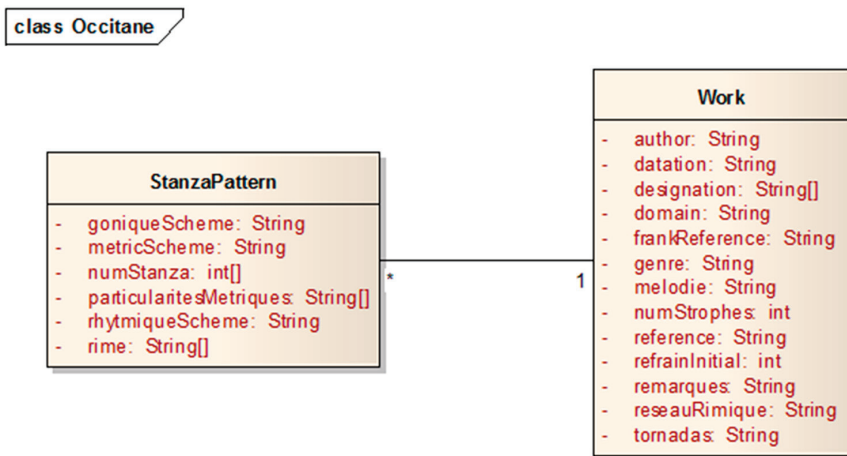
The fourth example presented here is the repertoire “Répertoire métrique de la poésie lyrique occitane des troubadours à leurs héritiers (xiii^e-xv^e siècles).” It is a local repertoire implemented in a worksheet, in one single sheet. This file has the following columns: Domaine, Datation, Auteur, Référence, Schéma Rimique, Schéma métrique, Référence Frank, Désignation, Genre, Refrain initiale, Nbre Couplets, Réseau rimique, Tornadas, Rimes, Particularités, Mélodie, Remarques. This is apparently a flat model but some of the cells have more than one value and refer other cells. This means that there are two concepts on the sheet: “Work” (a poem) and “StanzaPattern” (the way a poem is classified). We can define more than one type of stanza pattern in the same poem.

The UML Class diagram of the conceptual model of the local repertoire “Répertoire métrique de la poésie lyrique occitane des troubadours à leurs héritiers (xiii^e-xv^e siècles)” is shown in Figure 13.

Since this repertoire is a local Excel file there were no functional requirements to identify. The stylesheet has one only sheet with columns identifying properties, and lines, where each line is an instance of the concept and each cell has data of that instance related to the property of the column. No functionalities were identified.

All this work done with the Dimev, Versologie and Occitane databases is also documented in stylesheets. Each concept has its own list of properties, the process is the same as with the relational databases but since there are no tables, the process is slightly different. In the XML cases, elements are listed hierarchically and described, each element corresponds to a property. Properties are grouped by concept. In the

Figure 13. Conceptual Model of the “Répertoire métrique de la poésie lyrique occitane des troubadours à leurs héritiers (xiii^e-xv^e siècles)” local repertoire



Using Reverse Engineering to Define a Domain Model

case of the perl file, each concept has also a list of properties defined in the file, and in the case of the excel file, all properties are listed since the model is very flat, but grouped by the correspondent concept (there are only two concepts).

The information that comes from all the conceptual models, together with the information that comes from S1, POSTDATA is building a common conceptual model of the European Poetry. It is important to integrate all the dimensions of all repertoires and also integrate other sources as already mentioned so POSTDATA can include as many concepts as possible in the common model.

RECOMMENDATIONS

To develop a metadata application profile is a complex task due to the uncertainty of the possible uses of the data. This specificity enhances the fact that:

- A method should be used to help to control the complexity
- The more information that can serve as input for the development of the domain model, the better

In fact, it is impossible to know all the uses the final users will give to the data. By “final users” the authors mean users that will manipulate software that uses the data, or organizations or persons that develop software that uses the available data. The more information is used as starting point for the modeling the more are the chances to define a good model. It means that many “realities” or “views” of the context will be represented in the model.

According to Me4MAP, the S1 activity (developing the Functional Requirements) and the S2 activity (Developing the Domain Model) should be done one after the other (S2 after S1) as presented in Figure 16 of Appendix. Nonetheless, in the context of POSTDATA the analysis of the Webpages of the repertoires was performed at the same time as the analysis of the databases and development of the conceptual models of those databases. In this case both S1 and S2 activities resort to existent documentation or to Web interfaces owned by the community that will use the MAP. The POSTDATA team performed the S1 activity at the same time as some activities of S2

The order in which activities S1 and S2 are performed depends on the type of resources the work-team has as starting point to do the development. Generalizing, when the development of a Domain Model (S2) is based in existent documentation or systems’ analysis, this development can start at the same time as S1 since S2 is not entirely dependent on the results of S1.

Using Reverse Engineering to Define a Domain Model

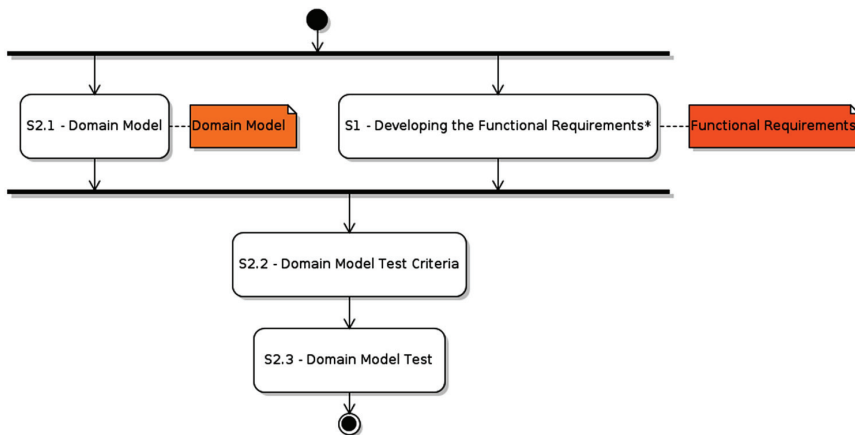
Me4MAP defines the need to develop the activity “High Level Requirements.” In the case of POSTDATA this activity is not yet explored, a reflexion has to be made in order to understand if there are more requirements than the ones identified due to the fact that this process is done in the context of the Semantic Web.

The POSTDATA team recommends though two changes to Me4MAP:

- Allow the S1 activity to be developed at the same time the development of the Domain Model (see Figure 14) in cases the MAP work-team resorts to existent documentation or analysis of existent systems to define the Domain Model,
- Discuss the possibility of defining the activity “Elicitation of high level requirements” as a non mandatory activity when the community of practice that will use the MAP has systems that can be the basis for functional requirements modeling.

The idea of developing a MAP using different sources such as MySQL dumps, XSD files, Perl files and stylesheets illustrates the versatility of the reverse engineering approach. A MAP development can resort to many knowledge already established in the software engineering community, since the early phases of a MAP development (domain modelling) have similarities with the early phases of software development.

Figure 14. Suggestion for new order of the activities S1 and S2 in Me4MAP



CONCLUSION AND FUTURE WORK

Poetry metrics measures and classifies poems counting syllables, accents, rhythm and rhymes to define the essential elements of the poem structure, its musicality and the type of contents that it shapes. A digital poetry metrics repertoire is a tool that gives account of metrical and rhythmical schemes of either a poetical tradition or school gathering a long corpus of poems, which are defined and classified by their main characteristics. There are many digital poetry metrics repertoires representing all European traditions, most of them in open-access in the Web of Documents. They are developed with different technologies and using different data models. Linked open Data technologies brought the possibility to open the silos of information that all these repertoires are and solve the interoperability issue that exists. For that, there is the need to define a common semantic model for all repertoires in order for them to be able to publish its data as structured Linked Open Data and become interoperable among all.

This common semantic model is a metadata application profile (MAP). The European Research Council (ERC) Starting Grant project POSTDATA is developing a MAP for the European Poetry following a method (Me4MAP) proposed recently by Curado Malta and Baptista (2013a). This chapter focus in the early stages of development of this MAP, and it shows in a very practical way what processes and techniques the POSTDATA team used to define the Functional Requirements and the Domain Model. This activities use as starting point the digital repertoires available on the Web of Documents, and a local repertoire. The modelling of the Functional Requirements resort to the analyses of the Websites by means of a reverse engineering process using the technique of use-case modelling. The definition of the Domain Model resorted to the analysis of documentation of the logical models of the databases, also by means of a reverse engineering process, to develop all the conceptual models of the repertoires. These conceptual models together with the Functional Requirements modelling and also information collected from:

1. Two Case Studies of researchers that are building a local repertoire at the same time the MAP development is in place
2. A survey that was done to final users, will be use to define the Domain Model

The POSTDATA team will continue with the activities defined by Me4MAP to develop the MAP for the European Poetry. In the near future, the testing of the Domain Model activity will be in place. This testing will be done with a Focus Group with representatives from all digital repertoires that were used as basis for the Domain Model development. After that the next main step will be the definition

of the semantic model, that is, which vocabularies and terms will be used and what constraints should be applied to the elements of the model.

POSTDATA expects to have the MAP for European Poetry ready by the end of 2017.

ACKNOWLEDGMENT

The authors would like to thank all the repertoires' responsible for their availability in sharing information and discussing database issues with the POSTDATA team.

Mariana Curado Malta thanks the Polytechnic of Oporto for her 3 years' leave which opened the possibility to work in POSTDATA, a wonderful and challenging professional experience in Madrid-UNED.

This paper has been developed thanks to the research projects funded by MINECO and led by Elena González-Blanco: Acción Europa Investiga EUIN2013-50630: Repertorio Digital de Poesía Europea (DIREPO) and FFI2014-57961-R. Laboratorio de Innovación en Humanidades Digitales: Edición Digital, Datos Enlazados y Entorno Virtual de Investigación para el trabajo en humanidades, and the Starting Grant research project: Poetry Standardization and Linked Open Data:POSTDATA (ERC-2015-STG-679528), funded by European Research Council (ERC) under the European Union's Horizon2020 research and innovation programme (<http://postdata.linhd.es/>).

REFERENCES

- Antonelli, R. (1984). *Repertorio metrico della scuola poetica siciliana*. Palermo: Centro di Studi Filologici e Linguistici Siciliani.
- Bianchi, A., Caivano, D., Marengo, V., & Visaggio, G. (2003). Iterative reengineering of legacy systems. *IEEE Transactions on Software Engineering*, 29(3), 225–241. doi:10.1109/TSE.2003.1183932
- Bouillon, L., Vanderdonckt, J., & Chow, K. C. (2004). Flexible Re-engineering of Web Sites. *Proceedings of the 9th International Conference on Intelligent User Interfaces* (pp. 132–139). New York, NY, USA: ACM
- Brunner, H., Wachinger, B. & Klesatschke, E. (2007). *Repertorium der Sangsprüche und Meisterlieder des 12. bis 18. Jahrhunderts*, Tübingen, Niemeyer.

Using Reverse Engineering to Define a Domain Model

Curado Malta, M., & Baptista, A. A. (2013a). A method for the development of Dublin Core Application Profiles (Me4DCAP V0.2): detailed description. In M. Foulonneau & K. Eckert (Eds.), *Proc. Int'l Conf. on Dublin Core and Metadata Applications 2013* (pp. 90–103). Lisbon: Dublin Core Metadata Initiative. Retrieved from <http://dcevents.dublincore.org/IntConf/dc-2013/paper/view/178/81>

Curado Malta, M., & Baptista, A. A. (2013b). Me4DCAP V0. 1: A method for the development of Dublin Core Application Profiles. *Information Services & Use*, 33(2), 161–171. doi:10.3233/ISU-130706

Curado Malta, M., & Baptista, A. A. (2013c). State of the Art on Methodologies for the Development of Dublin Core Application Profiles. *International Journal of Metadata, Semantics and Ontologies*, 8(4), 332–341.

Frank, I. (1957). *Répertoire métrique de la poésie des troubadours*, Paris, H. Champion.

Gómez Bravo, A. M. (1999). *Repertorio métrico de la poesía cancioneril del siglo XV*. Alcalá de Henares: Universidad.

Gómez Redondo, F. (2001). *Artes Poéticas Medievales*. Madrid: Laberinto.

Gorni, G. (2008). *Repertorio metrico della canzone italiana dalle origini al Cinquecento (REMCI)*. Firenze: Franco Cesati.

Hassan, S., Qamar, U., Hassan, T., & Waqas, M. (2015). Software Reverse Engineering to Requirement Engineering for Evolution of Legacy System. *Proceedings of the 2015 5th International Conference on IT Convergence and Security (ICITCS)* (pp. 1–4). IEEE. doi:10.1109/ICITCS.2015.7293021

Jacobson, I., Booch, G., Rumbaugh, J., Rumbaugh, J., & Booch, G. (1999). *The unified software development process* (Vol. 1). Addison-wesley Reading.

Linden, F., Schmid, K., & Rommes, E. (2007). The Product Line Engineering Approach. *Software Product Lines in Action*, 3–20.

Mölk, U., & Friedrich Wolfzettel (1072). *Répertoire métrique de la poésie lyrique française des origines à 1350*, Munchen, W. Fink Verlag.

Moretti, F. (2005). *Graphs, Maps and Tress – Abstracts models for a literature theory*. London: Verso.

Müller, H. A., Jahnke, J. H., Smith, D. B., Storey, M.-A., Tilley, S. R., & Wong, K. (2000). Reverse Engineering: A Roadmap. *Proceedings of the Conference on The Future of Software Engineering* (pp. 47–60). New York, NY, USA: ACM.

- Naetebus, G. (1891). *Die Nicht-Lyrischen Strophenformen Des Altfranzösischen. Ein Verzeichnis Zusammengestellt Und Erläutert*. Leipzig: S. Hirzel.
- Nilsson, M., Baker, T., & Johnston, P. (2008). The Singapore Framework for Dublin Core Application Profiles. Retrieved from <http://dublincore.org/documents/singapore-framework/>
- Nilsson, M., Baker, T., & Johnston, P. (2009). Interoperability Levels for Dublin Core Metadata. Retrieved from <http://dublincore.org/documents/interoperability-levels/>
- Pagnotta, L. (1995). *Repertorio metrico della ballata italiana*. Milano, Napoli: Ricciardi.
- Parramon i Blasco, J. (1992). *Repertori mètric de la poesia catalana medieval*. Barcelona: Curial.
- Pillet, A., & Carstens, H. (1933). *Bibliographie der Troubadours*. Halle: M. Niemeyer.
- Pohl, K., Böckle, G., & van Der Linden, F. J. (2005). *Software product line engineering: foundations, principles and techniques*. Springer Science & Business Media. doi:10.1007/3-540-28901-1
- Pressman, R. S. (2015). *Software engineering: a practitioner's approach* (8th ed.). Palgrave Macmillan.
- Raynaud, G. (1884). Bibliographie des chansonniers français des xiii. [treizième] et xiv. [quatorzième] siècles, Paris, Vieweg.
- Rumbaugh, J., Jacobson, I., & Booch, G. (2004). *Unified Modeling Language Reference Manual, The*. Pearson Higher Education.
- Sneed, H. M. (1995). Planning the reengineering of legacy systems. *IEEE Software*, 12(1), 24–34. doi:10.1109/52.363168
- Solimena, A. (2000). Repertorio metrico dei poeti siculo-toscani. Centro di studi filologici e linguistici siciliani in Palermo, Palermo.
- Somé, S. S. (2009). A Meta-Model for Textual Use Case Description. *Journal of Object Technology*, 8(7), 87–106. doi:10.5381/jot.2009.8.7.a2
- Tavani, G. (1967). *Repertorio metrico della lingua galego-portoghese*, Roma, Edizioni dell'Ateneo.
- Touber, A. H. (1975). *Deutsche Strophenformen des Mittelalters*. Stuttgart: Metzler.
- W3C (2015). Semantic Web. Retrieved from <https://www.w3.org/standards/semanticweb/>

Using Reverse Engineering to Define a Domain Model

Zenari, M. (1999). *Repertorio metrico dei 'Rerum vulgarium fragmenta' di Francesco Petrarca*. Padova: Antenore.

KEY TERMS AND DEFINITIONS

Conceptual Model: Or conceptual data model expresses the knowledge of an organization through concepts and relations between concepts.

Functionality: A functionality is the set of functions executed by the system that transform input data in output data in a way that the user obtains a useful result.

Logical data model: A data model of a specific domain that expresses data in terms of tables with columns and relations between tables, or XML tags or object-oriented classes.

Poetry Metrics: Poetry metrics measures and classifies poems counting syllables, accents, rhythm and rhymes to define the essential elements of the poem structure, its musicality and the type of contents that it shapes.

Relational Model: A way of managing data that was defined by Codd in 1970 in the seminal document "A Relational Model of Data for Large Shared Data Banks" (see <http://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf>).

Reverse Engineering Process: A backwards process of extracting knowledge or design from anything man-made.

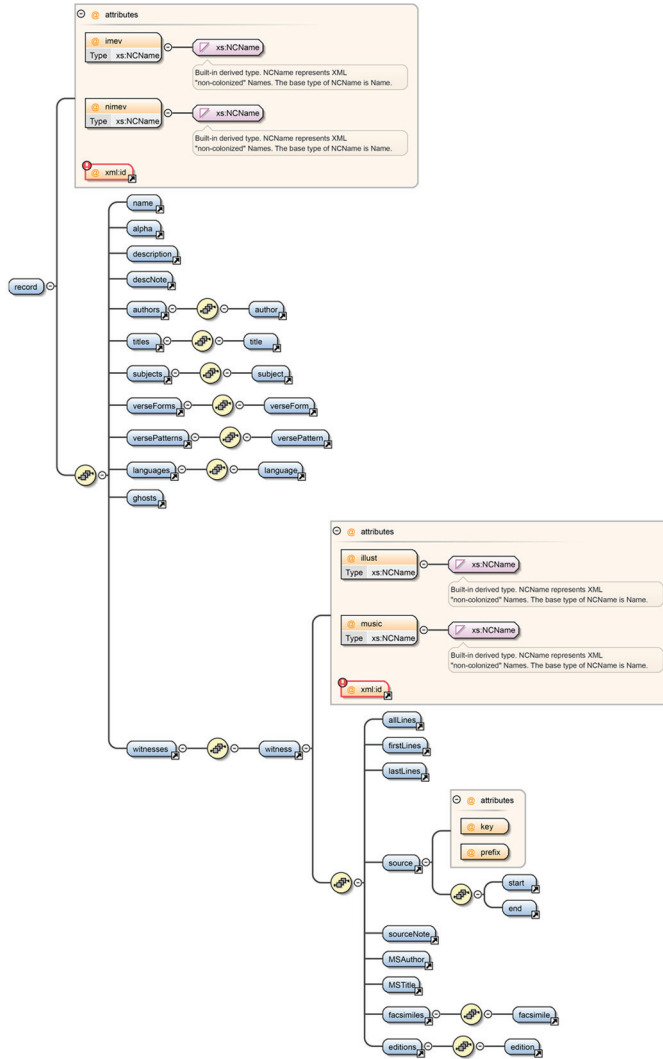
ENDNOTES

- ¹ See <http://rpha.elte.hu/>
- ² See <http://www.cirp.es/bdo/med/meddb.html>
- ³ See http://www.bedt.it/BEdT_04_25/inf_home_crediti.aspx
- ⁴ See <http://www.nouveaunaetebus.elte.hu>
- ⁵ See <http://csm.mml.ox.ac.uk>
- ⁶ See http://webserver.erwin-rauner.de/crophius/Analecta_conspectus.htm (not available in open access)
- ⁷ See <http://www.liederenbank.nl/>
- ⁸ See <http://www.corimu.unisi.it>
- ⁹ See <http://www.mirabileweb.it>
- ¹⁰ See <http://dimev.net>
- ¹¹ See <http://icalia.es/troubadours/ca/>
- ¹² See <http://www.versologie.cz/en>
- ¹³ See <http://www.remetca.uned.es>

- ¹⁴ See Manifesto for the Digital Humanities from <http://tcp.hypotheses.org/411>); Mathew Gold's Debates in the Digital Humanities, A Companion to Digital Humanities from <http://dhdebates.gc.cuny.edu/>
- ¹⁵ "Web of Documents" is a term used in contrast with the term Web of Data. The Web of Documents is made of documents read by human beings that navigate between documents located in servers through hyper-links, it is the Web that everyone uses in a daily basis. The Web of Data or Linked Data or even the Semantic Web, three ways of expressing similar concepts, have technologies that "enable people to create data stores on the Web, build vocabularies, and write rules for handling data" (W3C, 2015)
- ¹⁶ Only draft versions are published so far. The first version of Me4MAP was submitted to an international research journal and is waiting for approval. The POSTDATA team is using this first version of Me4MAP not yet published
- ¹⁷ See <http://www.oxfordlearnersdictionaries.com/definition/english/functionality?q=functionality>
- ¹⁸ See <http://www.remetca.uned.es>
- ¹⁹ See <http://www.mysql.com/products/workbench/>
- ²⁰ See <http://dimev.net>
- ²¹ See <https://www.oxygenxml.com/>
- ²² Text Encoding Initiative, see <http://www.tei-c.org>
- ²³ See <http://www.tei-c.org/release/doc/tei-p5-doc/en/html/VE.html>
- ²⁴ See <http://metro.ucl.cas.cz/kveta/>

APPENDIX

Figure 15. Digital Edition of the index of Middle English Verse XML schema



Using Reverse Engineering to Define a Domain Model

Figure 16. Me4MAP order of activities S1 and S2

