A Software Product Line for Digital Libraries

Delfina Ramos-Vidal, Alejandro Cortiñas, Miguel R. Luaces,
Oscar Pedreira and Angeles Saavedra Places
Universidade da Coruña, Centro de Investigación CITIC, Laboratorio de Bases de Datos,
Facultade de Informática, Campus de Elviña s/n, 15071, A Coruña, Spain

Keywords: Software Product Lines Engineering, Digital Libraries.

Abstract: Digital libraries (DL) have become very popular nowadays. The most important libraries of the world currently publish their collections online, properly classified and, in many cases, with access to the digital version of the resources. DLs are very useful for the general public and for research groups in many fields, such as language and literature. Developing these products is a complex and expensive task, and usually small organizations have to conform with DLs with poor features or outdated. However, this kind of systems usually share most of the features, and even the data model, maybe with little modifications depending on the kind of resources to manage. In this paper, we analyse the domain of DLs using both the previous knowledge of the research group and reviewing many existing products. The outcome is an exhaustive feature model that models the variability of this kind of systems.

1 INTRODUCTION

In the last decades, there have been many efforts to create *digital libraries* (DL) to manage and put in the web digital resources (and their bibliographic information), either obtained from digitization processes of literature and cultural heritage or native digital resources. Non-profit organizations that own valuable books, journals and pamphlets have seen the DLs as a means to preserve and disseminate their heritage. Research groups in Digital Humanities have created DLs as a base to foster research that would be unfeasible without their navigation and search capabilities.

The Database Lab, the authors' research group, has been involved in a number of successful multidisciplinary projects of digital libraries, in collaboration with experts in different areas of Humanities (Language and literature, Arts, Pedagogy, etc.). In the next section, we describe a selection of them that are representative in their areas.

The development of a DL is a complex process in many ways. Experts from Software Engineering and Humanities have to find a common language to understand each other. As a result of years of experience, we can conclude that the development of a DL from scratch is too expensive both in time and budget that can hardly be faced by non-profit organizations or research groups. However, there are a set of characteristics that DLs have always in common and hence a shared core of assets can be defined. Furthermore, even though each DL has its own particularities in terms of data model and characteristics, they can be analysed and defined using a formal language. Therefore, the domain of DL is suitable for the application of software product lines engineering (SPLE).

In this paper, we present the analysis of the domain of DLs following a specific methodology for the application of SPLE, and we define a complete feature model for DLs using the knowledge acquired by our research group. Hence, this paper constitutes a first step to achieve the generation of DLs automatically from a simple specification of its requirements. We also present an application example on how a particular DL can be defined from the feature model representing the variability of this domain.

The rest of the article is structured as follows: In Section 2 we present background and related work. In Section 3 we describe the domain and address the first stage of the SPL definition, which includes product planning and variability modelling. In Section 4 we present a case of use of the SPL, describing a real application by means of features. Finally, Section 5 presents the conclusions of the paper and lines for future work.

Ramos-Vidal, D., Cortiñas, A., Luaces, M., Pedreira, O. and Places, A.
A Software Product Line for Digital Libraries.
DOI: 10.5220/0010214903810389
In *Proceedings of the 16th International Conference on Web Information Systems and Technologies (WEBIST 2020)*, pages 381-389
ISBN: 978-989-758-478-7

2 BACKGROUND AND RELATED WORK

2.1 Digital Libraries

The **Library of Congress**[1] is the largest DL in the world, with millions of resources. It stores not only books or printed material, but also photos, drawings, newspapers, music, maps, and many other kinds of resources. Its web page is updated frequently with new collections and news. It provides a simple search engine to search for simple keywords, but the results can be filtered and sorted by many properties. The user interface is very friendly and modern, and the resources can be downloaded freely in different formats. This DL is used by the congress of the US as its main documentation provider.

Project Gutenberg[2] is a huge DL that provides over 60,000 free eBooks for online reading or downloading, mostly of old works for which U.S. copyright has expired. The DL is maintained by volunteers who digitize the content. Its interface is not very friendly, and its search engine capabilities are limited (it uses external search engines to search within the content of the works), but it provides categorized browsing from more general categories, or *bookshelves*, as the web calls them, to specific domains. The eBooks are also provided in different formats to facilitate the readers its download and posterior reading.

Cervantes Virtual Library[3] is one of most important digital libraries in Spain. Its catalogue is classified into different areas, from *literature* and *language* to *american library* or *galician literature library*. They support digital resources of many types: pdf, web, images and eBooks. Also, they provide metadata in MARC21 format (Library of Congress - Network Development and MARC Standards Office, 2016), the physical libraries standard. The search engine allows searching keywords in the content or the metadata, and it allows filtering depending on the kind of document.

Golden Age Digital Library (BIDISO)[4] is a portal that includes in fact many DLs related to the Spanish Golden Age, with collections of four different kinds: *historical emblems*, *news pamphlets*, *erudition resources*, and *inventories and libraries*. The web page provides a search engine that can query all collections simultaneously but also links to each one of

Figure 1: Screenshot of a resource webpage from Symbola.

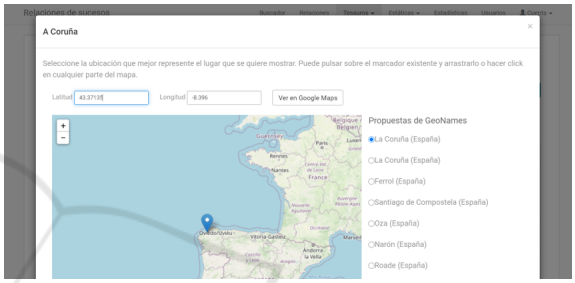

Figure 2: Screenshot of location management from CBDRS.

the related DL so the user can browse through its content. The user interface varies depending on the DL, being some of them very old and without modern and responsive interfaces, but other ones are quite modern and well design. Some of the DLs related are SYMBOLA or CBDRS. The former, **SYMBOLA**[5], is a DL specialized in historical devices or *impreses* (e.g., jousting devices, heraldic badges), from whom stands out its administration site management, including their feature of offering static pages to model additional information about the library itself, and the user interface friendliness and styling (see Figure 1). The latter, **Catalogue and Digital Library of News Pamphlets (CBDRS)**[6], is a DL of news pamphlets, where their advanced search feature, enabling exhaustive and restrictive filtering of the data, stands out, together with the geographical location management of publication places, and the inclusion of maps for its physical representation , visible in Figure 2.

The project **Critical Edition**[7], based on the study

[1]Library of Congress: https://www.loc.gov/

[2]Project Gutenberg: http://www.gutenberg.org/

[3]Biblioteca Virtual Miguel de Cervantes: http://www.cervantesvirtual.com/

[4]Biblioteca Digital Siglo de Oro (BIDISO): https://www.bidiso.es/index.htm

[5]SYMBOLA Divisas o empresas históricas: https://www.bidiso.es/Symbola/

[6]Catálogo y Biblioteca Digital de Relaciones de Sucesos (CBDRS) https://www.bidiso.es/CBDRS/

[7]Edición en Galiza durante a etapa franquista: http://ediciongalizafranquista.udc.gal/

of the publishing industry in Galicia during the Francoist period, was also taken into account. Protrudes their intricate data model, an ambitious approach that takes into account all different types of literary publications, them being editions, works, parts of works, collections of editions, and modelling the different relationships that can happen among them, as well as their authorities, which can sign each work with a different pseudonym, and the organizations in charge of executing the edition of each work. In this DL, the collections do not have digital resources, but external links to other DLs where the resources are located.

The project from the University of A Coruña, **Galician Virtual Library (BVG)** [8], the *Galician Virtual Library*, is an old fashioned web site with a large collection of Galician publications, with digital resources in different formats: text, audio, video, and images. It allows search by content using a strategy based on Bounded Natural Language. The resources stored in the database can have comments made by the users. It also handles editions from the same author, even when written under different pseudonyms.

2.2 Software Product Lines

Software Product Lines Engineering (SPLE) is a field of Software Engineering that focuses on reusing the same software artefacts among different software systems from the same family trying to reduce development costs and time to market such systems. This is, when a software development team has to develop a set of products that share most characteristics, both functional and non-functional, instead of carrying out the development of each one of them from scratch, it is efficient to develop a Software Product Line (SPL) and approach the development of all of them together (Pohl et al., 2005). This way, the base code of all the products is the same, and each of the actual products to sell or finally publish includes the share of this code that it needs depending on the actual functionalities it needs to provide. For example, if our company specializes on e-commerce web applications, some clients may need to have "tags" on the products, while others do not. Then, "having tags on the products of the shop" can be a variant feature that can be selected or not for a specific product. The source code this functionality is implemented will only be included in the final product source code if the feature is selected.

Designing and developing an SPL is a costly task. As we express in the above paragraph, it is more efficient than developing each product in a one-by-one basis, but however is also a more complex process

[8]Biblioteca Virtual Galega: http://bvg.udc.es

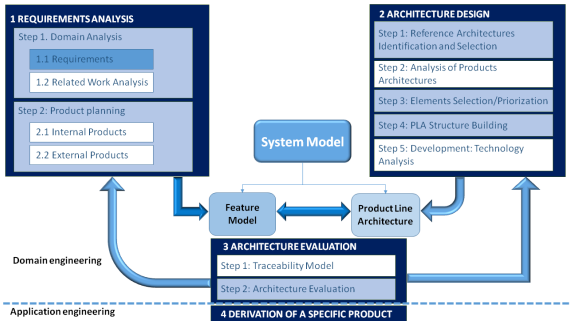

Figure 3: Methodology for constructing a SPL (Cortiñas et al., 2017).

since the decisions made affect not only to a particular product but to the whole set of them. There are different methodologies to follow when designing a SPL. The authors of this work defined one in a previous work (Cortiñas et al., 2017), suitable when the development team has access to both the source code of existing products of the family and to experts in the domain. This methodology divides the definition process in four stages, as seen in Figure 3: *requirements analysis*, *architecture design*, *architecture evaluation*, and *derivation of a specific product*. *Requirements analysis* stage is divided into two steps, *domain analysis*, which consists in analyzing both the requirements and the related work of the domain, and *product planning*, which consists on studying existing applications of the domain or family, including products developed from the team who is designing the SPL and products external to it. *Architecture design* focuses on studying relevant product architectures for the domain and prioritizing components, as well as for deciding the technology stack for the products. The output of both these stages is the input of the next one, *architecture evaluation*, whose purpose is to validate the models previously defined and to link the different assets defined in them to support the future evolution of the SPL. The last stage is the *derivation of a specific product*, which in fact takes part on a different context, the *application engineering*, this is, the team in charge of configuring a specific product is the one in charge of this product, not the whole SPL.

This work focuses on the first stage of the methodology, *requirements analysis*, with the objective of identifying the set of features that the whole possible products of a family could require in an exhaustive way. The output of this stage is an artefact called *feature model*, a tree diagram used to represent features of a product or family of products. A feature is a characteristic of a system relevant for some stakeholder, functional (e.g., a concrete requirement such as "user authentication") or not (e.g., the database management system to use) (Apel et al., 2013).

3 REQUIREMENTS ANALYSIS

This section illustrates the first stage for the definition of a software product line (SPL) in the domain of digital libraries following the methodology defined in (Cortiñas et al., 2017). First of all, we describe the domain. Then, we detail the two steps of this stage (see Figure 3). Finally, we show and explain the feature model defined as the result of the analysis.

3.1 Description of the Domain

DLs are information systems where large collections of information are stored in digital formats and offered to the users through one or many interfaces. This generic definition serves to give an impression of how wide is the spectrum of information systems that fit this domain. Usually, these collections of documents follow the traditional approach for publications. This is, there are authors, who write works, published in particular editions. DLs provide an extensive number of functionalities to manage the collections each one handle, and the different types of elements that are related such as editions of a work, collection of editions, parts of works, or standalone works, as well as the authorities involved, let them be writers, editors, printers or illustrators, to name a few. There are DLs that only handle metadata, but most of them also store digital resources linked to the elements of their collections, such as eBooks, pdfs, digitized pages or external links to other libraries.

In order to allow the users to search or navigate through the information, the metadata of the elements stored in a DL needs to be extracted and conveniently classified. Usually, DLs are managed collaborative, so there is a set of expert users in charge of handling the elements and their metadata, and there may be an internal reviewing process among different team members to ensure that the data updated to the library is accurate and of quality. This way, there are different roles for the users, and there may be variable types of registration for them, including registration through social networks. Of course, there are DLs without any users but a unique administrator who manages everything.

The navigation of the web application can be done through listings or by querying the database. The lists, forms and detail views of DLs are much more complex than the standard in web applications since there is a lot of information from related elements involved in, for example, an edition. Usually, searches area also quite complex in DLs since the standard user is normally trying to find something very concrete. Therefore, the search engine should be able to filter by any of the attributes of the elements. For example, the user may want to find an edition of a work written by a specific author, published in a range of years, with some particular text within its content. At the same time, the user interface needs to be as friendly as possible, considering the profile of most users, specialized in literature. Therefore, the design of these web pages is highly intuitive and transparent.

OAI-PMH is a protocol to share metadata between information systems through a harvesting process done by a server, the harvester, to many clients, the data providers. It is used to make libraries interoperable, allowing small DLs to aggregate their metadata collections into a web portal. There are examples of these portals at different administration levels, such as Hispana and Europeana. To provide the data from a DL into one of these portals, called aggregators, the library needs to support one or many metadata formats. This is, the metadata for the stored collections needs to be provided following some convection. Of course, apart from that, the DLs need to implement the client-side of the mentioned protocol.

There are some DLs in which the geographic scope is important. To address this, some features from geographic information systems may be added. For example, the elements of the library can be shown on a map viewer, and the users consulting the library can do queries with filtering the results by some geographic boundaries.

As we can see, most of the characteristics of this libraries are quite common, but there is some space for variability that depends, first of all, on type of the most important element the DL manages. For example, there are libraries whose collections are focused only on a particular author, like Valle-Inclán Digital Archive [9], and therefore in these libraries there is no point on showing and managing the set of authors; however, there are also libraries that study the authors of a particular period, such as the DL for the project *Publishing in Galicia during the Franco Era (1939-1975)*[10], and then the main page of the library should be the authors themselves, since they are the most important element of the library, and not their works. However, apart from this kind of variations, the attributes to record for the editions, or the authorities, are mostly the same among DLs, and the components to view or edit the information are very similar, except for the particular web style.

[9] Archivo Digital Valle-Inclán: https://www.archivodigitalvalleinclan.es/publica/principal.htm

[10] Edición en Galiza durante a etapa Franquista: http://ediciongalizafranquista.udc.gal/

3.2 Domain Analysis

We classified the requirements of this domain in four different groups: R1) Library objects, which includes all the requirements related to which kind of elements the library should handle (e.g., editions, collections, kinds of authorities, etc.); R2) Library data exportation, with the requirements regarding the ways the data may be exported from the library (e.g., supported formats, exportable elements, etc.); R3) User management, with requirements regarding the user roles the application should handle, and the features associated to each of these roles (e.g., reporters and collaborators, digital file access by anonymous users, the possibility of registering in the application, etc.; R4) Library access, which involves a huge set of variant requirements that involves the kind of navigation the DLs should support (e.g., the most important elements are editions, works or authors, the way the users can use search engines, etc.). Besides these, there is another requirement: the applications should be in a particular language, which for the DLs is always the same language of the content of the library.

We modelled the required features for systems of the domain of DLs with these requirements in mind. The result is 162 features that represent most of the relevant functionalities of these applications. Deciding the grade of the importance of each one of these features is done in the next step, *product planning*.

This list of requirements or features needs to be compared to other SPLs for DLs. To the best of our knowledge, there is not related work about software product lines applied on DLs. However, there have been some attempts to generate the software that implements products from this family using other techniques, so we can study these works as a reference.

Gonçalves et. al (Gonçalves and Fox, 2002) show a domain-specific language (DSL) based a formal model for DLs, that generates a DL application for a particular system, MARIAN. Using XML, they describe all the procedures and elements that the DL to generate requires, including, for example, the operations the system should handle, such as a search operation. This level of detail in the definition of the DL was required back in 2002, but nowadays software has evolved to a point where this is not required. Malizia et al. (Malizia et al., 2010) use a model-driven approach to generate DLs as Java desktop applications. Their models describe, as in the previous work, not only the elements of the library but also the operations to a high level of detail. It is difficult to compare the features these systems provide with our proposal since both of them are very flexible when defining a new product, so they can adapt better to fine-grained specific requirements of a DL. This flexibility comes at a great cost since the definition of a product in these systems is much more complex than a selection of features in a software product line.

Another interesting document is the DL reference model (Candela et al., 2008). The European Commission funded a Network of Excellence on DLs, called DELOS[11], whose main objectives were research and technology transfer, although it was abandoned in 2009. On their last period, DELOS was working on the development of a Digital Library Reference Model, designed to meet the needs of the next-generation systems and a globally integrated prototype implementation of a Digital Library Management System.

The research laboratory of the authors of this work, besides working on specific DL as mentioned in Section 2.1, have previously worked in projects targeting improvements on the software supporting these systems to simplify their implementation, always in collaboration with researchers from the Digital Humanities area. For example, one of the projects focused on enhancing the behaviour of compression and search algorithms when the text is written in a romance language and, in addition, publishing the algorithms and tools developed as open source. In another project, major goals were the development of advanced tools to facilitate the establishment of DLs on the web, while improving the quality of their services from a multidisciplinary point of view. In this last project, two specific tools were developed: a first tool for rapid prototyping of DLs, and a tool to feed these DLs with XML extracted from documents.

3.3 Product Planning

The purpose of this step is to verify the requirements extracted from the previous step with actual products in production. Besides that, using these products we can establish which features are common to all the family of products, which are variant, what kind of relationship exist between a feature and its sub-features, and also to determine the constraints between the features that are not related with the feature tree itself. Last goal of this step is to priority the features in order of importance, so the implementation of the base code for the SPL can be useful to develop basic DLs in early stages, and the development afterwards can take advantage of the users of these initial products, improving the quality of the final interfaces and functionalities with their feedback.

[11]DELOS Network of Excellence: http://delosw.isti.cnr.it/

As products that our laboratory has developed in the past, and therefore we have access to the software assets, data model, etc., we have decided to use four products presented in Section 2.1: *SYMBOLA*, *CBDRS*, *Critical Edition* and *BVG*. These four products are representatives examples which differ in their nature, features, domain, and in their time of creation. For example, *BVG* and *Critical Edition* have been developed long time ago, while *CBDRS* and *SYMBOLA* actual versions are very modern, having responsive interfaces to access from mobile devices. Another point of variability is the way the content is thought to be accessed. *BVG* and *SYMBOLA* designs are focused on the navigation through categories and listings, and particularly in the former there is not a search functionality. *Critical Edition* and *CBDRS* main entry are advanced search engines with many filters and features. There are more points of interest about these applications, such as the complete and complex data model in which *Critical Edition* data is stored, or the geographic features of *CBDRS*.

In order to confirm the completeness of the analysis, we need to compare our set of requirements with external products. We decided to use the *Cervantes Virtual Library* as one of the more complete DLs that we know.

3.4 Feature Model

The resulting feature model is shown in Figure 4. Each requirement group described in Section 3.2 derived in a high-level feature: `LibraryObjects`, `LibraryDataExportation`, `UserManagement`, `LibraryAccess`, and `Language`.

`LibraryObjects` feature (35 sub-features) is directly related to the kind of data the DL supports. `LO_E_ElementTypes` sub-features includes a feature to enable works, editions, parts of works and collections (first two are mandatory since we have not found a DL without this kind of elements). There is also a feature to enable the capability of managing the locations where editions are published, including setting the actual geographical coordinates in a map or managing older naming for the locations. `LO_E_SchedulingPublication` enables scheduling elements for publication, instead of showing them the moment they are created. `LO_E_DigitalResource` includes sub-features to decide which kind of digital resources are allowed. `LO_Authority` includes sub-features to define the possible roles of the authorities managed, the possibility for authors to have aliases or pseudonyms, and to store contact information for the authors. `LO_Organization`, which is optional, enables managing the organizations, and it has sub-features to decide if the director role is stored.

`LibraryDataExportation` feature (17 sub-features) groups all the features related to exporting data from the listings of the DL. The sub-features define the possible formats of the exported information (pdf, excel, txt, etc.), where is the feature applied (single elements or multi-element exportation), and which kind of elements can be exported.

`UserManagement` feature (19 sub-features) enables the user authentication module, which is mandatory since in every DL is required at least one user with the administration role. There are features to enable other three roles: reporter, which is the standard registered user role; collaborators, which is the people that collaboratively digitizes the content; and anonymous user. Other relevant features are related to each role, such as allowing anonymous users to access the digital resources or requiring the review of an administration in order to publish some content. Allowing using social networks account for authentication is also another feature provided, as well as allowing the anonymous users to register in the DL as reporters.

`LibraryAccess` feature (82 sub-features) is composed by every characteristic of the DL related to its user interface. It has two main sub-features, `PublicAccess` and `ManagementAccess`, which encompass the features for the anonymous users and reporters, and for collaborators and administrators, respectively. The two sub-trees are equivalent, describing the lists and search engines that the DL provides. The public interface differentiates between two possible search engines: a simple one, which is accessed through the menu bar from all across the DL, and an advanced one which has its own specific link, and which has options to enable searching by different properties. Besides that, the listings of the DL can have different types of pagination style (depending on the number of elements expected), and also they can be standard listings or grid views with thumbnails of the elements. Management interface related features are more simple, since there is only an advanced search engine, with specific options such as filter elements not reviewed, or filter elements which are still drafts, for example. Besides that, `PublicAccess` has some sub-features that define which is the home page of the DL, which can be a list of a particular element kind (editions, authorities or works), the list of latest publications, or a static page that can be managed by the administrator (if the feature related to this capability is selected).

`Language` feature (3 sub-features) serves to select the language in which the generated application user interface labels should be. For the libraries anal-

ysed, three languages need support: English, Spanish and Galician.

Besides the 162 features, there are 13 cross-tree constraints that affect 22 features. Some of these constraints are natural, since for example the homepage of the public access cannot be the lists of editions if listings are disabled (`PA_N_HP_Editions implies PA_N_L_Editions`, or *"DL public homepage is the list of editions" implies "DL provides the lists of editions for public access"*).

4 RUNNING EXAMPLE

In order to validate our proposal, we have selected an additional DL that was developed by our group and we have tried to define it by means of our feature model.

Poliantea[12] is one of the DL part of the BIDISO portal. It handles works published between XVI and XVIII centuries that focus on encyclopedic knowledge. Its target audience are students of literary, iconographic and cultural production for the mentioned centuries, since having these works classified and easily accessible is very helpful. The last update of the web application was on 2011, so definitely could use an upgrade to a version with modern technologies. Next, we describe the characteristics of the DL, and we reference the features that support each of them.

The main page of the library is the list of the editions (`PA_N_HP_Editions`). There is only one user who has the administration role (`UM_UR_Admin`), so there is no need for any other kind of users. However, anonymous users can access the digital resources provided by the DL (`UM_UR_AnonymousCanAccessElementFiles`). Administration navigation is very simple, just a list with all the editions (`PA_N_L_Editions` and `PA_N_L_E_AsList`), without pagination (around 140 editions stored currently).

Regarding the public access navigation, the list of editions does have, in this case, simple pagination (`PA_N_L_P_E_StandardPagination`). Apart from that, the user has two different search engines. The basic one searches by title (`PA_S_S_ByTitle`), author (`PA_S_S_ByAuthor`), year of publication (`PA_S_S_ByYear`), and by lasts resources added to the DL. The latter is the one characteristic of the DL that is not supported by the set of identified features for our SPL, since it was not defined as a common requirement for this family of products in our analysis. The advanced search engine can search by the same properties, but also by location (`PA_S_A_ByLocation`) or by date range instead of a single date (`PA_S_A_ByDateRange`). Finally, the search results are the editions that match the query, and they are shown in a list (`PA_SR_Edition` and `PA_SR_E_F_AsList`).

The main element stored in the DL is, as we can see for the characteristics mentioned so far, the *edition* (`LO_E_ET_Edition`), which in the case of Poliantea can be a publication of a particular work (`LO_E_ET_Work`), which is the standard, or to be modelled with more detail and therefore composed by parts of works (`LO_E_ET_Part`). The elements can have two kinds of digital resources: pdfs (`LO_E_DR_PDF`) and digitized pages (`LO_E_DR_DigitalizedPages`) (which are collections (`LO_E_DR_Collections`) of scanned images of the work). This way, an edition can be composed of several parts, each one with its own digitized pages. The editions stored in the library have an author (`LO_A_RoleAuthor`), and an editorial (`LO_O_RoleEditorial`) (which in the current version is a simple text field).

Summing up, the Poliantea DL can be defined by means of features of our proposal, using specifically 55 features, which are the ones mentioned and the higher-level features of the tree.

5 CONCLUSIONS AND FUTURE WORK

In this work, the first steps for the definition of a software product line in the field of digital libraries are addressed. We carried out an exhaustive analysis of existent DLs, extracting common requirements and identifying the set of features that support this family of products. Afterwards, we established the relationships between these features, and we identify which ones are part of every product of the family, and which ones belong to the variability of the product line.

The work presented in this paper is far from finished. According to the followed methodology, the next stage is to design the adequate architecture for the product line, using reference models or architectures for proved successful DLs. After that, a validation process will occur in which the decisions made so far are checked, and the two previous stages may need to run again to make small adjustments for the consistency of the whole design. Finally, the SPL needs to be implemented, from the product generation mechanism to the different components that provide for the

[12]Poliantea: Enciclopedias, repertorios de lugares comunes y misceláneas de erudición humanística: https://www.bidiso.es/Poliantea/

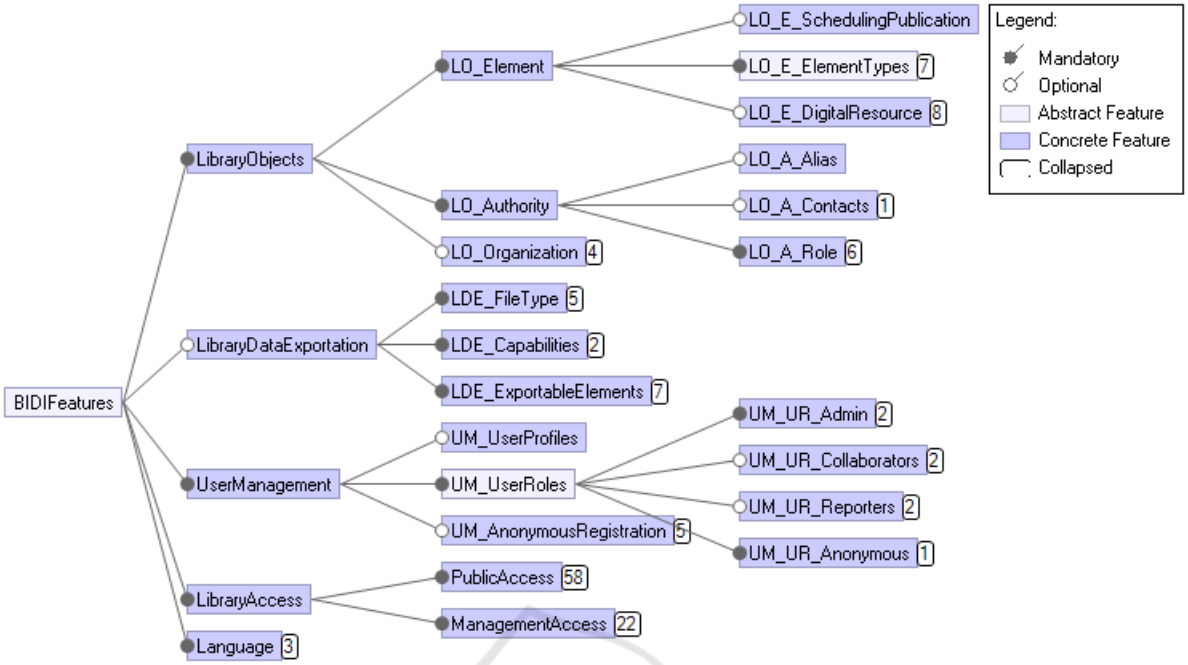

Figure 4: Simplified feature model (many features are collapsed).

features of the product family.

One of the main difficulties that will be managed during the development of the SPL is that many features directly affect the data model of the products, and implementing that kind of variability changes is a challenge.

The final validation of the SPL will be carried out by using it to generate new versions of the DLs from the Aracne Network[13], a platform that joins many DLs from the Spanish territory in a joint effort, encouraging new initiatives to improve the communication, convergence and projection of the work in the field of Digital Humanities.

ACKNOWLEDGEMENTS

Partially funded by: MICIU/FEDER-UE, MAGIST: PID2019-105221RB-C41; Xunta de Galicia/-GAIN Innovapeme: IN848D-2017-2350417; Xunta de Galicia/FEDER-UE, ConectaPeme, GEMA: IN852A 2018/14; Xunta de Galicia/FEDER-UE GRC: ED431C 2017/58; MICIU/FEDER-UE STEPS: RTC-2017-5908-7; MICIU/FEDER-UE BIZDEVOPSGLOBAL (RTI-2018-098309-B-C32); MINECO-AEI/FEDER-UE Datos 4.0: TIN2016-78011-C4-1-R; MINECO-AEI/FEDER-UE Velocity: TIN2016-77158-C4-3-R; CITIC research center funded by XUNTA and EU through the European Regional Development Fund- Galicia 2014-2020 Program, grant ED431G 2019/01.

[13]Red Aracne: http://www.red-aracne.es/

REFERENCES

Apel, S., Batory, D., Kästner, C., and Saake, G. (2013). *Feature-Oriented Software Product Lines*. Springer.

Candela, L., Castelli, D., Ferro, N., Koutrika, G., Meghini, C., Pagano, P., Ross, S., Soergel, D., Agosti, M., and Dobreva, M. (2008). The delos digital library reference model. foundations for digital libraries (version 0.98).

Cortiñas, A., Luaces, M. R., Pedreira, O., Places, Á. S., and Pérez, J. (2017). Web-based geographic information systems sple: Domain analysis and experience report. In *Proceedings of the 21st International Systems and Software Product Line Conference-Volume A*, pages 190–194. ACM.

Gonçalves, M. A. and Fox, E. A. (2002). 5sl: a language for declarative specification and generation of digital libraries. In *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 263–272.

Library of Congress - Network Development and MARC Standards Office (2016). Marc 21 format for bibliographic data. https://www.loc.gov/marc/bibliographic/. visited on 2020-09-21.

Malizia, A., Bottoni, P., and Levialdi, S. (2010). Generating collaborative systems for digital libraries: A model-driven approach. *Information Technology and libraries*, 29(4):171–186.

Pohl, K., Böckle, G., and Linden, F. V. D. (2005). *Software Product Line Engineering: foundations, principles and techniques*, volume 49. Springer Science & Business Media.