Oddrun Pauline Ohren

The DELOS
Digital Library Reference Model as a basis for describing and evaluating search systems

Masteroppgave
Avdeling for journalistikk, bibliotek og informasjonsfag

Summary

In this thesis I have studied the DELOS Digital Library Reference Model (DLRM) in detail, with the aim to evaluate its suitability as a description profile for search systems, - the ultimate purpose being to apply such descriptions as a basis for comparison and evaluation of search systems.

Two different approaches has been taken to evaluate DLRM. Firstly, a description profile – DLRM Vocabulary – has been derived from the original DLRM (as represented in Protegé) and applied to describe the end-user-accessible functionality in Web of Knowledge, WorldCat Local at State Library of Ohio and Europeana. Secondly, an analytical evaluation of the DLRM Vocabulary as a modeling language has been performed according to an existing quality framework for modeling languages.

The main conclusion is that DLRM Vocabulary does have potential as a description profile for digital libraries. No major inconsistencies or conflicts in DLRM Vocabulary are discovered, and the analytical evaluation indicated that it is fairly comprehensible, at least for audience with some knowledge on enterprise modeling. The main problem with using it in its present state is lack of specificity in some areas, causing construct overload or construct deficit.

Masteroppgave ved Høgskolen i Oslo, Avdeling for journalistikk, bibliotek- og informasjonsfag
Oslo 2009

CONTENTS

List of figures

Preface

My motivation for this work has been an interest for evaluation and analysis together with a profound liking for conceptual modelling. Lucky then that DELOS Network of Excellence came out with their rich reference model of digital libraries, with which I have spent many hours of study.

Many thanks to my supervisor, Ragnar Nordlie for helping slice the work down to something manageable, and for valuable input to approach and process.

My employer is worth many grateful thoughts. Tone Moseid and Leikny Haga Indergaard at ABM-utvikling have shown great generosity and flexibility concerning work hours and conditions in general.

Thanks also to my family for their firm support all the way, in spirit and in practice.

But first and last thanks are due to my husband, Stian, for his encouragement and help throughout, and especially during the last weeks, without which I would not have made it.

This work was conducted using the Protégé resource, which is supported by grant LM007885 from the United States National Library of Medicine.

Oslo 25. June 2009

Oddrun Pauline Ohren

1 Introduction

In this day and age with the ubiquitous, increasingly comprehensive internet/web, advanced retrieval services become more and more important. In the library world computerised information retrieval systems have been around for several decades. During the 1990ies, web search engines made their appearance, initially considered separate species from IR systems and library catalogues. During the recent years, however, it is more and more the case that any collection "has to be" searchable through the web; any IR system, any library catalogue has a web interface. Hence, the distinction between traditional IR systems, catalogue indexes and web search systems is getting blurred.

Evaluating search systems has proved to be notoriously hard, but, with the plethora of search services to choose from – increasingly important. The existing approaches vary greatly, from the system-oriented IR tradition in which precision and recall (and variants thereof) still are the dominant performance indicators, to the HCI approach, in which usability and perceived usefulness are the most important evaluation factors, see 2.2 for an overview.

In 2004 DELOS was established as a network of excellence[1] in Digital Libraries under the umbrella of EU framework programme 6 – Information Society Technology, with the aim to *"bridging the gap between this vision and the reality, by furthering research in many critical aspects of digital libraries and by the creation of an active European digital library research community."* (Speiser, 2007). The DELOS community has made important contributions to the research on digital libraries, one of the most important being the DELOS Digital Library Reference Model (DLRM) (DELOS Network of Excellence on Digital libaries, 2007), a formal conceptualisation of the digital library domain.

1.1 Approach and problem statement

One element of evaluation is *comparison*, either between the objects to be evaluated, between an object to be evaluated and some benchmark or standard, or between two states of the same object described at different points in time. When the objects to evaluate are intangibles, like in our case, we need them to be represented or described in a preferrably uniform manner. We make the assumption that DLRM, as a comprehensive model of the digital library

[1] Network of Excellence are instruments in the EU framework programme 6 – Information Society Technology (2002-2006)

domain, contains the concepts and relations necessary to describe any aspect of a digital library.

In this context I want to explore the use of DLRM as basis for a description profile for digital libraries.

More specifically, my aim is to

1. Investigate how digital libraries/search systems can be described by means of DLRM;
2. Investigate how and whether the descriptions created on the basis of DLRM incorporate the important and distinguishing features of the search systems, in such a way that comparisons are possible.

Thus, this thesis is about evaluation on two levels. Evaluation of search systems provide the context and ultimate purpose of the work reported herein, while the work itself is in effect an evaluation of the suitability of DLRM as a description device for search systems.

1.2 Method

To get as much insight as possible into DLRM's potential as description device, I chose to combine empirical testing with analytical evaluation, through the following process:

1. **Develop a description profile from DLRM.**

A description profile based on DLRM was developed. The profile is based on the existing concepts/classes in DLRM. Some of the concepts are enriched with necessary attributes to express information not possible to represent as relations to other concepts. Moreover, the description profile is specified according to the DCMI Abstract Model (Dublin Core Metadata Initiative, 2007). The description profile is hereby referred to as DLRM Vocabulary.

Currently the DELOS DLRM is published as a textual document containing descriptions and visualisations from a set of graphical models. Each concept is only represented by a textual definition and relations to other concepts, no other attributes are given. Therefore, to provide a better basis for overview, analysis and use of DLRM I found it necessary to transfer it into a computer readable form. For this an ontology editor, Protegé (Protegé, 2009) was chosen, rich enough in expressive power to represent all the construct types in DLRM. However, since the support for structured ontology reports from Protegé is limited, I had to use Excel and Visual Basic to create some useful DLRM overviews

2. **Test-of-concept: Apply DLRM Vocabulary on real search systems.**

Three search systems were selected for the test. The selected systems have different yet overlapping target groups, and are all accessible through a web interface. Between them they cover important types of the search systems, each with their own specialty:

- Web of Knowledge (Thomson Reuters) gives access to scientific publication and citation information;
- WorldCat Local (State Library of Ohio) is a library catalogue system with many advanced facilities;
- Europana (Europeana Thematic Network) gives access to cultural heritage material.

All three systems have been described using the DLRM Vocabulary, focusing on the *functionality* from the end user perspective as it may be accessed through their web interface.

Finally the expressivity and discrimination power of the DLRM Vocabulary were analysed based on experience gathered during the description effort and the descriptions themselves.

3. **Analytical evaluation of DLRM Vocabulary.**

To complement the test in 2, which focused only on the functionality domain in DLRM, an analytical evaluation of DLRM Vocabulary as a whole was performed, using a quality framework for modelling languages (J. Krogstie, 2008).

4. **Summing up findings.**

Finally, the findings from empirical test and analytical evaluation are analysed to form a unified judgement on whether DLRM has any potential role in evaluation of search systems.

1.3 The structure of this thesis

Chapter 1 (this chapter) gives an introduction, states the problem and outlines the approach.

Chapter 2 gives a context by outlining the research on evaluation within three related areas: Information retrieval, human computer interaction and digital libraries.

Chapter 3 describes how the DLRM vocabulary was derived from the original DLRM, including the principles that were followed and challenges met.

Chapter 4 describes and analyses the test-of-concept.

Chapter 5 describes the analytical evaluation of DLRM Vocabulary.

Chapter 6 contains discussion of findings, conclusion and suggestions to further work.

Appendix 1 describes the encoding of DLRM into Protegé.

Appendix 2 gives an overview of the classes and connected properties in DLRM, in the form of an Excel table.

Appendix 3 contains the full DLRM Vocabulary specification.

Appendix 4 contains a summary of the properties in DLRM Vocabulary, grouped according to their domains.

2 Evaluation of search systems – methodologies and approaches

Since information systems began to be developed, evaluation has been an important issue. The motivations/goals for individual evaluation efforts are manifold and include as diverse things as: to justify funding (to development project), to identify and explore new qualities of information systems, to verify the fulfilments of a priori requirements, to give input to return-of-investment analysis, to test the system against established standards, to compare the systems to similar systems (e.g. with the aim to select one among several alternative systems), and more.

In an Oxford dictionary (Hornby, 1994) evaluation is defined as "*find out or form an idea of the amount or value of something*". Obviously the concept of *value* is crucial in evaluation, yet very few authors in the body of literature relevant for this thesis discuss this fundamental issue. The reason for this may be that in the various research areas the concept of *value* long since has been translated or transformed into more concrete and easily assessable quantities like performance, usability, robustness, capacity, etc, each reflecting aspects of value.

However, Saracevic and Kantor specifically discuss the value concept in a study concerned with the value of library and information services (T. Saracevic & Kantor, 1997a, 1997b). They distinguish clearly between the value of *information* and the value of *information services*, and argue that the value of information services can be discussed and assessed independently from that of the information provided by the service. They suggest a model for *information use* reflecting the activities involved in using information: i) acquistion (getting the information) ii) cognition (absorbing, understanding the information) and iii) application (use the newly understood information for something). Based on this a model for *information service use* was proposed, the Reasons-Interaction-Results (R-I-R) model: i) Reasons for use (providing context), ii) Interaction with the service iii) Results from the interaction. The R-I-R model is then meant to represent dimensions of the value of an information service. Thus, according to (T. Saracevic & Kantor, 1997a, 1997b), evaluation of an information service can be seen as an assessment (by users) of the quality of their interaction with the service, and the benefit of the results from the interaction, - all in the context of their reasons for using the service in the first place.

Several research areas are relevant regarding search system evaluation, - notably information retrieval, digital libraries and human computer interaction (HCI). In the subsequent sections research related to evaluation within each of the three areas is outlined.

2.1 Evaluation of information retrieval systems

Evaluation in the research area of information retrieval stems from the Cranfield experiments ((Cleverdon, Mills, & Keen, 1966) cited in (Harman & Voorhees, 2006)) designed in 1960ies. Guided by the Cranfield model TREC (Text retrieval Conference) started in 1992, providing test collections and infrastructure for testing IR systems and prototypes (Harman & Voorhees, 2006). Through the years TREC has expanded and diversified into branches incorporating things like retrieval in other languages than English, web retrieval, interactive retrieval (including end users), retrieval of multimedia content, and other.

The basic elements in TREC are large test collections of documents, a collection of queries and relevance assessments. The tests are performed by running the queries against the document collection, without end user involvement and in a highly controlled setting, using quantitative indicators like precision and recall as measurements. A methodological contribution to IR evaluation was brought forward by Tague-Sutcliffe (Tague-Sutcliffe, 1992), in which she presents 10 decisions that have to be made when an evaluation effort is considered. The decisions in effect comprises an evaluation process handbook, from *Decision 1: To test or not to test?* to *Decision 10: How to present results*. While many of her principles and points are generally applicable in any evaluation project, the process is still firmly grounded in the controlled experiment tradition.

Traditional IR is typically summative in nature, in the sense that the systems to be evaluated are run through standard tests resulting in some quantitative score. Moreover, evaluation is based on the assumptions that relevance of a resource can be decided based on the document and query alone, and that the information need of the user does not change during an information seeking session. This has been criticised by several researchers. Hider (Hider, 2006) observes that users' search goals are not static throughout a search session, but may change as a result of new information gained in the process. Turpin and Scholer (Turpin & Scholer, 2006) detected in their study a significant lack of correlation between user judgement and the performance measures of the system. In his model of information behaviour (Pharo, 2004) (part of the Search Situation Transition method schema (Pharo & Jarvelin, 2004)) Pharo states that *"The search task's (like the work task's) may be vague at the start of a* *search and may be sharpened through interaction with the system."* Ellis (Ellis, 1996) is critical to the very foundation of IR evaluation, and argues that the field is based on wrong premises:

> *"…,information retrieval research is a discipline which is inextricably grounded in cognition.*
> *In this respect, the problems of measurement are more similar to those of psychology than those of physics. The archetypal and probabilistic approaches do not escape this cognitive foundation in that the success of the system is assessed against criteria, in form of relevance*
> *judgments, which are implicitly cognitive, but they treat the cognitive element of the research design, the relevance judgments, as being different from what they are in reality."*

The research in IR evaluation is still to a large extent based on classic experimentation, and the repeatability and transferability of evaluations to other cases for comparison is a very strong goal. Nonetheless, there has been a growing recognition that retrieval research must take the end user into account. With increasingly advanced and diverse user interaction mechanisms in search system, the search algorithm is no longer the only indicator of quality. Various measures have been taken to evaluate IR systems in a user context rather than as a matter between the query and the IR algorithm.

Borlund (Borlund, 2003) has suggested an evaluation framework for IIR (interactive IR) as a means to retain experiment-like control of the variables, yet making the experiment more realistic. She criticises the Cranfield approach in which information need is treated as a static object which is entirely reflected by the query, and argues that formation of information need should be treated as a user-individual and dynamic concept. She tries to resolve the problem along two lines, by

- suggesting alternative ways of measuring relevance and performance, where the judgement of the end user influences the measure;
- introducing context to the search via *Simulated work task situations*, - a description of a work scenario in which the test person is asked to take part.

Borlund's framework was used in a study of Blomgren, Vallo and Byström (Blomgren, Vallo, & Byström, 2004). The conlusions concerning the evaluation framework was that the Simulated work task situations functioned well, but that the correspondence between user judgment and the alternative performance measurement was poor, and in general that qualitative measures are important in real life settings.

2.2 Evaluation in Human computer interaction (HCI)

HCI studies interaction between computers and people, see e.g. (Sharp, Rogers, & Preece, 2007) for comprehensive information about HCI. As a research field it is considered to be on the "soft side" of computer science, including disciplines like behavioural science and graphical design. This heterogeneity is also reflected by the research community, involving both computer scientists, psychologists and designers.

The field covers the whole life cycle of interactive products. Interactive design is about *"developing interactive products that are easy, effective and enjoyable to use – from the users' perspective"(Sharp et al., 2007).* It then follows, that *evaluation* in HCI aims to find out how easy the product is to use, whether the users like the product. Moreover, the typical HCI evaluation is *formative* (as opposed to summative), meaning that it is often performed as part of the development process, so as to point out improvement areas for the next version/further development.

Usability is an important concept in HCI, as indicated by a study by Xie (2006). One much used definition is "how efficiently and effectively users can achieve their goals with the system" (Blandford & Bichanan, 2003). According to ISO as cited in (Fuhr et al., 2007) "usability is the extent to which a product can be used by specified users to achieve specific goals with effectiveness, efficiency and satisfaction in a specified context of use." Originally branded by the HCI community, it has come to be synonymous with Jakob Nielsens 10 usability heuristics (Nielsen, 1994), focusing on the user interaction design.

During the last 5-10 years usability studies have gained some foothold in the digital library field, and is also introduced in IR.

Within the digital library research several authors have taken a very broad view of the usability concept, and almost use it synonymously with "quality". This all-inclusive notion of "usability" makes it somewhat difficult to distinguish between general evaluation studies and usability studies. Hence, Blandford and Buchanan (2003) use "usability" literally as "can be used" and have identified the following criteria for digital library usability: usability performance measures, learnability, error tolerance and recovery, user experience and context of use. Note that in this conceptualisation usability also includes performance issues, as measured by e.g. precision and recall.

Chowdhury, Landoni and Gibb have performed a review of usability studies of digital libraries (Chowdhury, Landoni, & Gibb, 2006). They report of a diversity of approaches and observe that there exist as yet no standard set of tools or benchmarks for digital library design and evaluation. However, generic guidelines suggested in (Blandford & Bichanan, 2003), (Tefko Saracevic, 2004) and others should be used as a basis for usability studies in the digital library domain. The review by Chowdhury et al also revealed that in addition to factors connected to user interface, design and other technical issues, there are a number of other usability factor related to digital libraries, like: globalisation and localisation, language (e.g. multilingual access), cultural and multicultural issues, content characteristics (e.g. mediatypes, heterogeneity) and human information behaviour issues. The authors conclude that

> *"Digital libraries are designed for specific users and to provide support for specific activities. Hence digital libraries should be evaluated in the context of their target users and specific applications and contexts."*

Within the IR area, Ahmed, McKnight and Oppenheim has developed a comprehensive methodology for user-centered design and evaluation of IR interfaces (2006). The methodology is described as a process in which each step provide input to the next, involving a number of methods: i) Usability testing of a competitor system, ii) User task analysis iii) Heuristic evaluation of prototype design iv) Formative evaluation of improved prototype design v) Summative comparative evaluation (comparison with competitor system).

Based on the experience with their methodology, the authors suggests 9 principles for interface design for IR systems, which might be considered as an adaption of Nielsen's list of usability principles.

2.3 Evaluation of digital libraries

The studies reporting digital library evaluation in the previous section had their focus on usability. Here we will look at research in digital library evaluation in general.

Digital libraries as a research field is relatively newer and had an explosive growth in the 1990ies (T. Saracevic, 2000). The research in this field has to a large degree been separated from the IR evaluation research, although digital libraries and IR systems have much in common.

The explosive growth in digital library research mentioned above does not apply to *evaluation* of digital libraries. This is also true for the branch of DELOS working with evaluation, in which less strong results were obtained. Nevertheless, there have been suggested some systematic approaches which are both comprehensive and generic, and seem promising.

However, DELOS has not been the only arena for doing evaluation research on digital libraries. Blandford et al (2008) focus on the *evaluation process* and propose a general evaluation framework in the form of six stages, which is subsequently applied to perform a series of case studies. The six stages are: i) Formulate purpose/goal of evaluation, ii) Identify resources and constraints iii) Identify any ethical considerations connected to the evaluation, iv) Identify and select data collection schemes, v) Select analysis techniques, vi) Report findings. Bertot et al (Bertot, Snead, Jaeger, & McClure, 2006) performed a multi-year study in which they concluded that to assess a digital library from a user perspective functionality, usability and accessibility must be evaluated.

In general, DL evaluation efforts typically take a holistic approach, aiming to evaluate not only user functionality or system performance measured by specific criteria, but all aspects of a digital library, and seen from various perspectives. One might say that the researchers have taken a step back and not leapt to the task of finding criteria and indicators.

This has lead to a more comprehensive and systematic approach, in which one considers that any evaluation must answer the questions of what (aspect or component of a digital library) to evaluate, why evaluate (what is the purpose/motivation/goal of the evaluation), which criteria reflect performance on the evaluation purpose and, and how to evaluate (how to measure the criteria, that is, methodology) (Tefko Saracevic, 2004). These four dimensions constitute a framework for DL evaluation, which was used as a basis for the evaluation models proposed by DELOS (Fuhr et al., 2007) Both models took a very generic view of digital libraries consisting of three main components, namely content, users and system/technology. The classification and evaluation scheme (Fuhr, Hansen, Mabe, Micsik, & Sølvberg, 2001) attached criteria to each main component, while the triptych model considered interaction between each of the components as the crucial point, see

Figure 1.

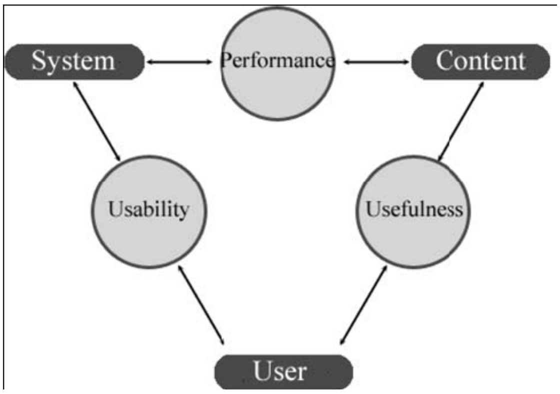


Figure 1 The interaction triptych model

In this model evaluation implies judging the quality of the interaction between each component pairs, e.g. between content and user.

Both models take the position that to evaluate and compare digital libraries in a holistic and comprehensive manner there must be a uniform way of describing them in reasonable detail. For this to make any sense it is of course vital that the description include the elements that are considered to be important features for digital libraries. The description scheme provided by DELOS is a simple list of attributes, and although the attributes do represent important features, the description scheme as such is not explicitly related to DLRM at all.

At this point, it is my opinion that the DELOS DLRM is the obvious source for generating such a description scheme ("metadata format") for digital libraries, and as already stated, in this project I will investigate more closely into how suitable DLRM is as a description language of search system functionality.

3 Developing the DLRM Vocabulary

In this chapter the foundation for a description profile for digital libraries will be made, and the development method described. DLRM Vocabulary itself is specified in Appendices 3 and 4.

3.1 Encoding DLRM into Protegé

To facilitate better overview and analysis of DLRM, its salient parts have been encoded into Protegé 4.0 (Protegé, 2009), an open source ontology editor based on OWL. Although mainly a straightforward task, the encoding process did pose some challenges, - some due to peculiarities in the DLRM, some caused by limitations in Protegé/OWL.

Below the most important principles used during encoding are outlined. Consult Appendix 1 for a more comprehensive explanation.

3.1.1 Concepts

Entities defined as *concepts* are encoded into Protégé as *classes*, organised hierarchically according to their <is-a> relation. Each class is enriched by up to 3 types of annotations:

- *label*: concept name
- *isDefinedBy*: the textual definition specified in DLRM
- *comment*: rationale and examples as specified in DLRM. Additional comments are included as deemed appropriate or necessary.

3.1.2 Relations

Entities defined as *relations* are encoded into Protégé as *properties*. Domain and Range are specified as classes. Their textual definition is recorded by the isDefinedBy annotation, rationale and examples as a Comment annotation. Additional annotations are included as deemed appropriate or necessary.

Ternary relations are transformed into 2 binary relations, see for example *associatedWith*.

3.1.3 Domains as concepts vs. organising constructs

In general, the concepts defined in DLRM are entites that may be instantiated into concrete objects in the digital library world, that is, digital libraries, digital library systems, digital library management systems, or component thereof. Example: The concept of Digital Library may be instantiated into a specific digital library, e.g. the ACM Digital Library.

Domains are another type of entities in DLRM. Although defined and described as “ordinary” concepts in the DLRM, they have a different meaning compared to the other concepts. Domains are constructs defined to impose structure on the reference model itself, meaning they are not to be instantiated into the real world of digital libraries, but to stay – and be used - at the model level.

However, since we wanted to represent the domains in the DLRM ontology, we found that the “cleanest” solution is to encode domains as classes. See Appendix 1 for discussion.

3.2 From DLRM to description format

There is no direct way from DLRM in its current form to a description format allowing for a flexible description of a digital library. While the DLRM concepts are richly connected by way of defined relationships which may be used to denote properties of the connected concepts, it is not obvious which kind of resources/concepts a description format actually should describe. According to DLRM the digital library conceptual universe contains nearly 240 interrelated concepts, many of which denote core types of resources in a digital library.

On this background several questions need to be discussed:

- Which of the resource types included in DLRM should be described by their own description format?
- Do the defined relationships constitute an appropriate set of properties to be recorded in the description format?
- Do we perhaps need additional properties connecting the digital library resources to external domains like time and space, or properties with literal values like textual descriptions?

One obvious place to look for input concerning these questions is the Dublin Core Metadata Initiative (Dublin Core Metadata Initiative).

3.3 Dublin Core Metadata Initiative

In the cultural heritage area Dublin Core is largely considered to be the common denominator between metadata schemes for archive objects, bibliographical objects and museum objects, respectively. We therefore wish our new description profile to be developed along the lines of Dublin Core. In our case, the following two specifications from Dublin Core Initiative are important:

- DCMI Metadata Terms (Dublin Core Metadata Initiative, 2008a) : The authoritative specification of all metadata terms maintained by the Dublin Core Metadata Initiative, including the 15 original elements (the terms of the Dublin Core Metadata Element Set), additional properties, classes for denoting domains and ranges for the properties, and the DCMI Type Vocabulary (Dublin Core Metadata Initiative, 2008b), a list of terms denoting general resource types;

- DCMI Abstract Model (Dublin Core Metadata Initiative, 2007): An abstract model for Dublin Core metadata, describing the components and constructs used. Hence, it is in effect a metamodel for DCMI Metadata Terms. The DCMI Abstract Model is particularly important for developers of metadata schemes which are based on DCMI Metadata Terms. Future adoption into DCMI terms is much easier if the expansion comply with the DCMI Abstract model.

DCMI Abstract Model prescribes a model for vocabularies (e.g. DCMI Metadata Terms) as shown in the UML model[2] in Figure 2.

[2] Those unfamiliar with the UML static diagram symbols should note that the *squares* denote *classes* or *entity types*, *interconnecting lines* (directed or undirected) denote *associations* between the linked classes and *lines with blocked arrowheads* denote *specialisation/generalisation* (can often be read as “is-a”, e.g. Class is-a Term)

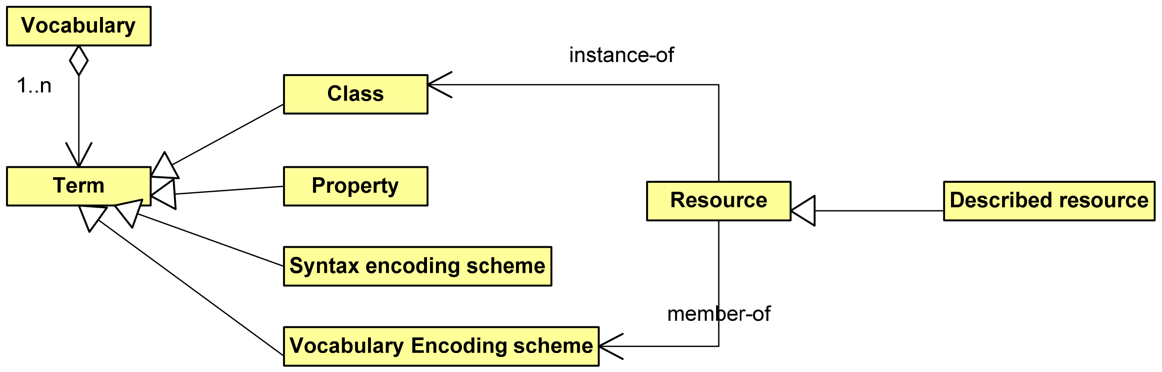


Figure 2 Vocabularies as modelled in DCMI Abstract Model

A vocabulary contains one or more terms. Each term may be a class, a property, a syntax encoding scheme or a vocabulary encoding scheme. Described resources may be instances of one of the classes in the vocabulary.

In the rest of this chapter we are going to specify our description profile – hereafter referred to as DLRM Vocabulary - according to the DCMI Abstract Model.

3.4 *Constructing DLRM Vocabulary according to DCMI Abstract Model*

At this point a few key concepts from DCMI Abstract Model should be defined:

1. A *description* is made up of one or more *statements*, all about one and the same resource (the one-to-one principle). The resource described by a description is referred to as the *described resource*.

2. Each statement represents a *property-value* pair (property-i, value-j), expressing that the described resource has the value *value-j* on the property *property-i*.

3. Several descriptions (each describing a single resource) may be grouped into *description sets*, typically because the described resources are related in some way. In our case an example could be a description set including one description of a digital library, one of its collection, one of the functions it offers, etc.

4. A *vocabulary encoding scheme* is an enumerated set of terms, typically to be assigned as value to some property. Example: DCMI Type Vocabulary is a set of terms from which values can be selected and assigned to the property dcterms:type.

Now we need to do the following:

1. Identify the concepts in DLRM to be represented as classes in DLRM Vocabulary (concepts that can be a "described resource"
2. For each resource type (class), specify which properties should be properties in DLRM Vocabulary. This also involves specifying value types (range), and in which cases vocabulary encoding schemes should be used.
3. Where appropriate define vocabulary encoding scheme and/or syntax encoding schemes for some of the properties

3.4.1 Identify classes in DLRM Vocabulary

Figure 3 is a UML model showing the core concepts in DLRM.

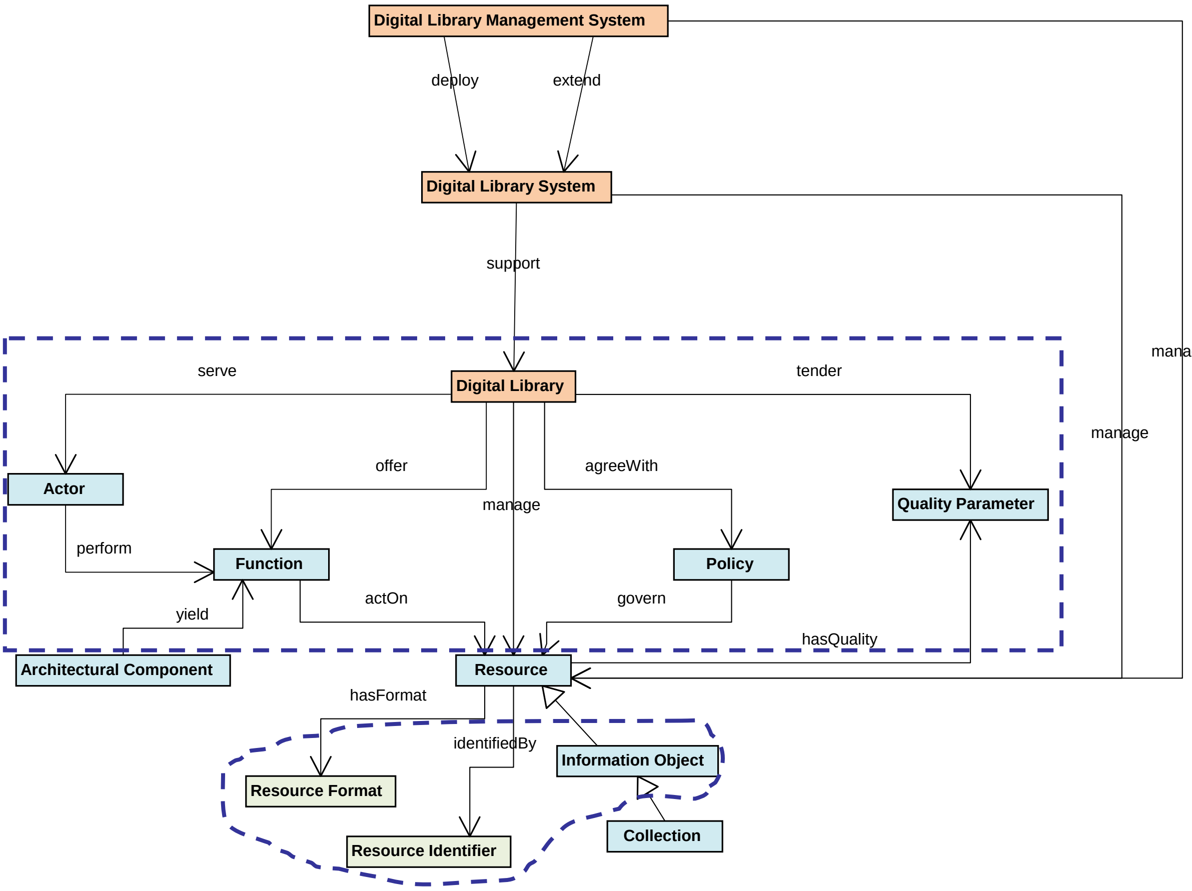


Figure 3 Core concepts in DLRM

In Figure 3 colour coding is used to enhance readability. The three system levels of the digital library universe are orange. The blue concepts are all different kinds of resources, whereas the rest are coloured pale green.

The key concept is Digital Library, related to the topmost concept in each of the DLRM domains in the following ways:

A *digital library*

- *manages* the *resources* (including the information objects and collections)
- *serves* one or more *actors*,
- *offers* one or more *functions* (*yielded* by *architectural components*)
- *agrees with* one or more *policies*
- is *supported* by a *digital library system,* which is again *deployed* by a *digital library management system*

Then, based on the concepts in Figure 3 and in compliance with the DCMI Abstract Model we could say that a description set for a digital library may contain

- a description of at least one *digital library*;
- descriptions of the *functions* offered by the digital library;
- descriptions of the *actors* served by the digital library;
- descriptions of the *policies* agreed with by the digital library;
- descriptions of the *information objects* (including collections) managed by the digital library;
- descriptions of the *quality parameters* tendered by the digital library;
- description of the *digital library system* supporting the digital library;
- description of the *digital library management system* by which the digital library system is; deployed
- descriptions of the *architectural components yielding* the functions of the digital library.

All the resource types indicated in the list above may represent *described resources* in a description set about digital libraries, it is not feasible here to specify new description formats for all of these types. The concepts *Architectural Component*, *Digital Library Systems* and *Digital Library Management System* all belong to the Software system domain, having computer oriented properties, many of which not particularly relevant for describing digital libraries[3].

Hence, the terms that are classes in DLRM Vocabulary are Digital Library, Information Object, Actor, Function, Policy, Quality Parameter, Resource Identifier, Resource Format (indicated with the dotted frames in Figure 3).

3.4.2 Identifying properties for the selected "described resources"

In this section the properties of the description formats of Digital Library, Function, Policy and Quality Parameter will be identified and defined[4]. Candidate properties for inclusion in the description format for any of the four concepts will be identified by performing the following steps:

1. Any relationship (defined in DLRM) *from* the concept in question (i.e. relationships for which the concept in question is the domain) is a candidate property. Example: In DLRM the relation influencedBy is defined from Function to Actor Profile. Hence, influencedBy is a candidate property for Function, having some Actor Profile as value.

2. Any relationship (defined in DLRM) *from* the more general concepts of the concept in question (i.e. relationships for which any of the more general concepts is the domain) is a candidate property. Example: In DLRM Resource is a generalisation of many other concepts, including Function, Policy and Quality Parameter. Hence, any relationships defined from Resource may be applicable as properties for Functions.

3. Any relationship (defined in DLRM) *from* any of the more specific concepts of the concept in question (i.e. relationships for which any of the more specific concepts is the domain) is a candidate property. Example: In DLRM the concept of Function is subdivided into many subtypes of functions. Any relationship defined *from* any of the

[3] This may seem strange, all the while e.g. WorldCat Local is an instance of a Digital Library System. But since the *function* concept is anchored in the digital library, not the digital library system, there is no gain here in using the latter as our object of study.

[4] To limit the scope, InformationObject and Actor will not be detailed further.

subtypes may be applicable as properties of Functions (to be used in the cases where the function to be described is of that particular type).

4. Properties from any of the Dublin Core Metadata Initiative (Dublin Core Metadata Initiative) specifications, of which the specifications mentioned in 3.3 seem particularly relevant. As DLRM is fairly comprehensive concerning relationships between concepts (meaning that properties with non-literal values are well covered) , the most likely candidates from the Dublin Core sources are properties with literal values, like description, title, etc.

3.4.2.1 Properties for digital libraries

Figure 3 visualises all relations defined for the concept of Digital Library in DLRM. Digital Library. Based on this the properties *serve*, *tender*, *agreeWith*, *manage* are added to the DLRM Vocabulary. So are *description*, *title* and *type* from DCMI Terms

3.4.2.2 Properties for all resources

In DLRM Resource is the abstract[5] superclass of both Function, Policy, Quality Parameter, Actor and Information Object. Hence all properties defined at Resource level may also be used as properties for the more specific resource types.

According to the model in Figure 4 the properties *associatedWith*, *hasQuality*, *regulatedBy*, *identifiedBy*, *hasFormat*, *belongTo*, *hasPart*, *describedBy*, *expressedBy*, *hasAnnotation* and *hasMetadata* can be added to the DLRM Vocabulary. In addition the properties *description*, *title* and *type* from DCMI Terms are made applicable for all resources.

[5] Classes declared as *abstract* can not be instantiated directly, only through its non-abstract subclasses. Hence, we may create an instance of Function (e.g. one instance may be the *Google I'm feeling lucky* function), but not an instance of Resource. Resource is purely conceptual, defined to encompass everything (properties, features) that is shared among all the resource types in a digital library.

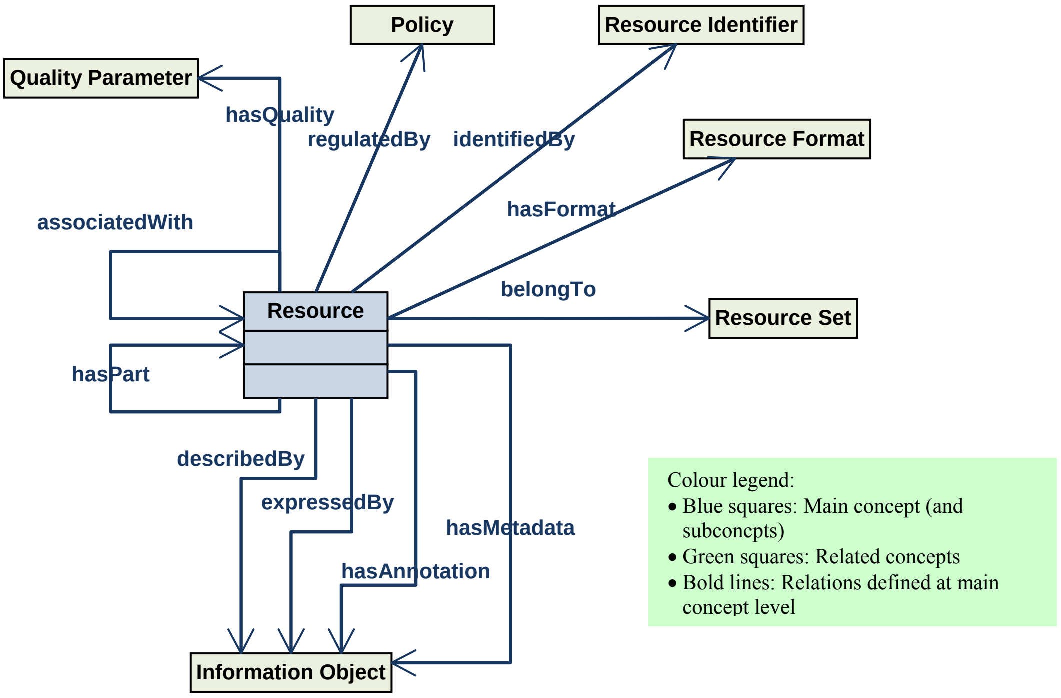


Figure 4 Properties of resources in general, as specified by relations in DLRM

3.4.2.3 Properties for Function

Figure 5 visualises a fragment of the DLRM Function domain. Only function types for which specific relations are defined are included in the diagram, together with their "is-a path" to the Function concept itself. Hence, Figure 5shows all relations that DLRM specifically defines for functions.

Thus, according to Figure 5 the properties *issue*, *return*, *retrieve*, *interactWith*, *actOn*, *influencedBy* and *create* should be added to the DLRM Vocabulary.

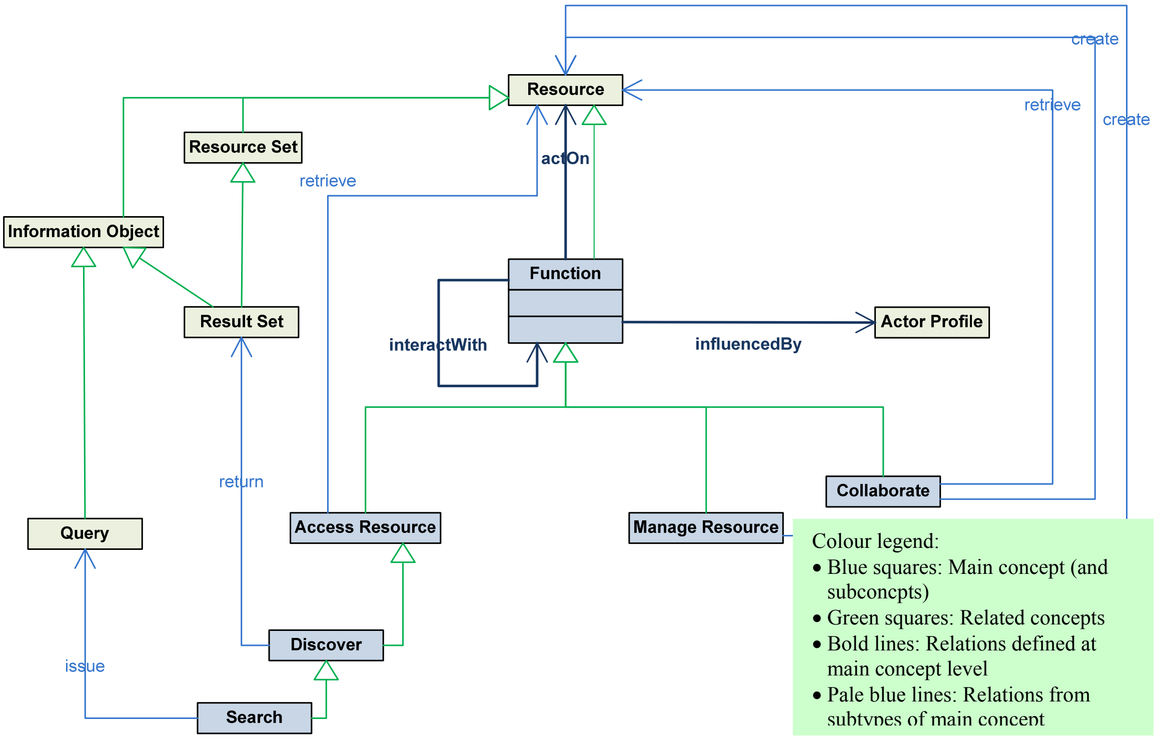


Figure 5 Properties of functions as specified by relations in DLRM

3.4.2.4 Properties for Policies

Figure 6 gives an overview of the relationships specifically defined for the Policy concept in DLRM.

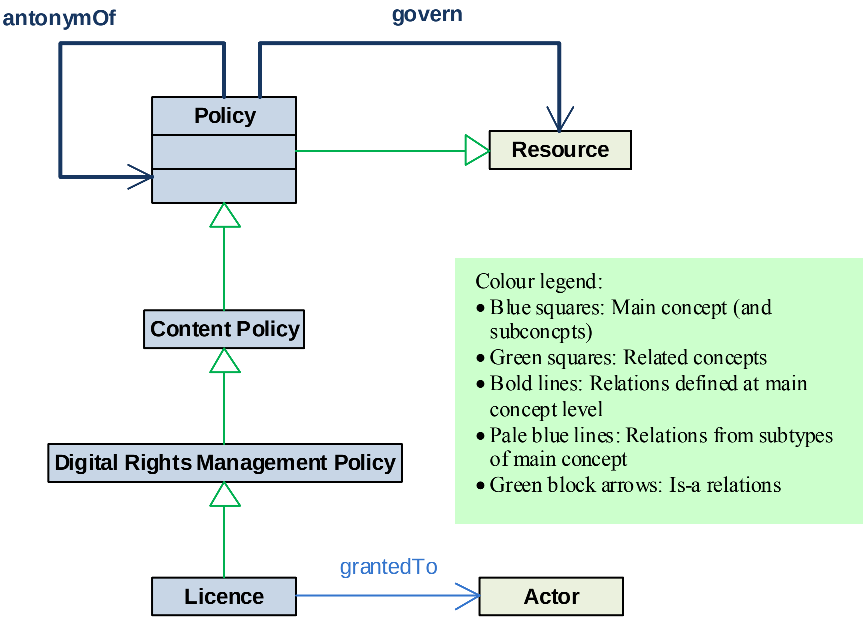

antonymOf
govern
Policy
Resource
Content Policy
Colour legend:
• Blue squares: Main concept (and subconcpts)
• Green squares: Related concepts
• Bold lines: Relations defined at main concept level
• Pale blue lines: Relations from subtypes of main concept
• Green block arrows: Is-a relations
Digital Rights Management Policy
grantedTo
Licence
Actor

Figure 6 Properties of policies as specified by relations in DLRM

Note: As I understand the relation antonymOf is a relation between policy *subclasses* (policy types), not between policies themselves. For instance, the Policy subclasses *Implicit Policy* and *Explicit Policy* are antonyms, so are *Voluntary Policy* and *Enforced Policy*. While antonymOf can not be included as a property of Policy, the *antonym subclass pairs* do get used as a basis for corresponding Boolean properties of Policy, e.g. isExplicit, isEnforced, etc.

Hence, the properties *govern*, *grantedTo*, *isExplicit*, *isEnforced*, *isExtrinsic* and *isPrescriptive* are added to the DLRM Vocabulary.

3.4.2.5 Properties for Quality Parameters

Figure 7 gives an overview of the relationships defined for the Quality Parameter concept in DLRM.

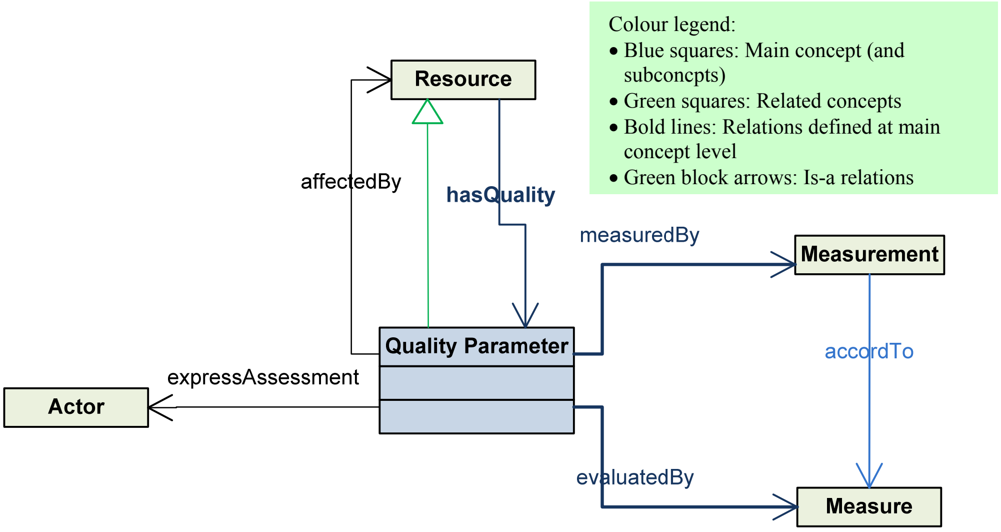

Colour legend:
• Blue squares: Main concept (and subconcpts)
• Green squares: Related concepts
• Bold lines: Relations defined at main concept level
• Green block arrows: Is-a relations
Resource
affectedBy
hasQuality
measuredBy
Measurement
accordTo
Quality Parameter
expressAssessment
Actor
evaluatedBy
Measure

Figure 7 Properties of quality parameters as specified by relations in DLRM

From the model above we get the properties *expressAssessment*, *affectedBy*, *hasQuality*, *measuredBy* and *evaluatedBy* as new properties to DLRM Vocabulary

To simplify the description profile *Measurement* is not a class in our test. Measurements are given as literal values where applicable

3.4.3 Vocabulary Encoding Schemes in DLRM Vocabulary

From 3.4.1 it should be clear that we have chosen to operate with quite general classes in our DLRM Vocabulary. To be able to retain some of the information represented by the subclass hierarchy of each of the vocabulary classes, type vocabularies are established for each class.

In addition a property <class>Type (<class> should be replaced by ‘function’, ‘policy’, etc) is added to each class, to store selected terms from the type vocabulary. For example, a search function in Europeana is instantiated as *Function*. To indicate that it is a search function it is assigned “Search” (from the vocabulary of function types) to its property *functionType*.

Altogether five vocabulary encoding schemes are established, the last one being a general type vocabulary, to be used to assigned value to the property type. All instances of the same class have the same value from this vocabulary, e.g. all functions have *type*=*Function.*

The vocabulary encoding schemes are specified and explained in Appendix 3.

4 Testing DLRM Vocabulary on real search systems

In this chapter we will try to test out the suitability of using DLRM Vocabulary for describing search systems, when the motivation for doing so is evaluation of those systems. The description profile extracted from DLRM covers all the main resource types "as DLRM sees it", and consists of

- A set of properties applicable for the various types of resources and for the digital library as a whole

- A set of type vocabularies to be used for some of the properties.

To be able to understand a phenomenon or object based on a description, it is vital that the description is representative, meaning the description must include the distinguishing features of the object. Moreover, to compare several objects or phenomena based on descriptions, the descriptions should support comparison by being structured the same way, and containing the same type of information. Using a common description profile will of course ensure this.

To get an indication of the suitability of the DLRM Vocabulary, we need to verify that the vocabulary terms (properties and type vocabulary terms) really points to features that matter in the various types of resources. Furthermore, the profile should be able to represent *all* the important features, and with a suitable granularity. In more generic terms, we could say that the main thing we need to find out about the DLRM description power is its *discrimination power* and its *expressivity*, which, to a certain extent correspond to the *domain appropriateness* in the SEQUEL model quality framework (J. Krogstie, 2003; J. Krogstie, Sindre, & Jorgensen, 2006).

As a test of the DLRM Vocabulary, the functions of three different search systems have been described according to the DLRM Vocabulary. The systems selected are Europeana, WorldCat Local (as implemented at State Library of Ohio) and Web of Knowledge. The test, including data registration and results are described in 4.3 onwards.

4.1 Test of expressivity

As a basis for judging the expressivity of the DLRM description profile, we need some independent definition or model of the things that are to be described. For information objects this might be taken for granted, - the long tradition of metadata creation and management has left us with a pretty clear idea what needs to be described about documents and other "regular" information objects, at least in a library context. This is not necessarily so for other, more intangible objects like functions, policies and quality parameters, nor for information objects like collections, ontologies and queries, all of which are *resources* according to DLRM.

For functions, which is the key concern of our test, there exist several definitions, more or less formal. In mathematics the idea of function is understood as a mapping from one set of objects (the domain) to another set of objects (the range) in such a way that to any given object of the domain the function assigns one and only one object in the range[6]. Similarly, in the context of computer systems functions are usually understood as operational units that – upon receiving zero or more input arguments , perform a series of steps and produce some output[7]. In our case we choose to generelize "output" to "result", including not only functions returning some value, but also functions that just lead to changes in some of the data structures of the system.

In our test, only functions available for the user are described as such, not the (sub)functions that may in turn be invoked internally by the user functions. In addition to input data and result, for users the interaction mechanism is an important feature of functions. Hence, for this test we adopt a notion of function that consists of four elements:

1. Input arguments or initial state
2. Altgorithm
3. Output or result
4. Interaction mechanism

The above being a very basic and generic "model" of functions, not taking into account any particularities of the search system domain, the descriptions should at least be able to cover this.

[6] Formally: A function is a set of pairs $(a_{i,} b_i)$, in which $a_i \neq a_j$ for any i and j
[7] Merriam-Webster: "a computer subroutine ; *specifically* **:** one that performs a calculation with variables provided by a program and supplies the program with a single result"

4.2 Test of discrimination power

To use descriptions of resources as basis for *comparisons* between the resources, the descriptions must fulfil the following requirements:

1. Descriptions of similar resources should be similar (R1 ≈ R2 ⇒ D(R1) ≈ D(R2))

2. Descriptions of different resources should be different (R1 ≠ R2 ⇒ D(R1) ≠ D(R2))

3. Similar descriptions should imply similar resources (D(R1) ≈ D(R2) ⇒ R1≈ R2)

The requirements above correspond to the notions of ontological completeness and clarity described in (Wand & Weber, 1993): construct redundancy (1), construct overload (2 and 3).

In 4.5 the extent to which the descriptions of the user functionality of Europeana, WorldCat Local and Web of Knowledge provide us with the means to identify important differences across systems will be discussed.

4.3 Data registration

As already pointed out, the test-of-concept was performed for functions only, mainly because the functionality in my opinion is the very core of a search system. Moreover, functions are accessible and investigable from the "outside", at least functions that can be invoked by the user.

Hence, only user initiated functions are described. When functions are related to other types of objects, like policies or information objects, these are also described, albeit not as detailed. Information objects, for instance, are not assigned values to properties other than those linking the information object to functions, or those linking the information objects to an object containing information that is vital to the test (most often *Resource Format*), or properties with literal values (*description*, *type*). Descriptions are encoded in Excel sheets, of which fragments are shown in Figure 9 to Figure 13. For each system there is one sheet for the functions, one for the related objects. In the function sheet each column corresponds to a property. Both properties specifically defined for functions and properties defined for resources in general are included. Functions and related objects were given a descriptive name, with a prefix showing the system in which they belonged:

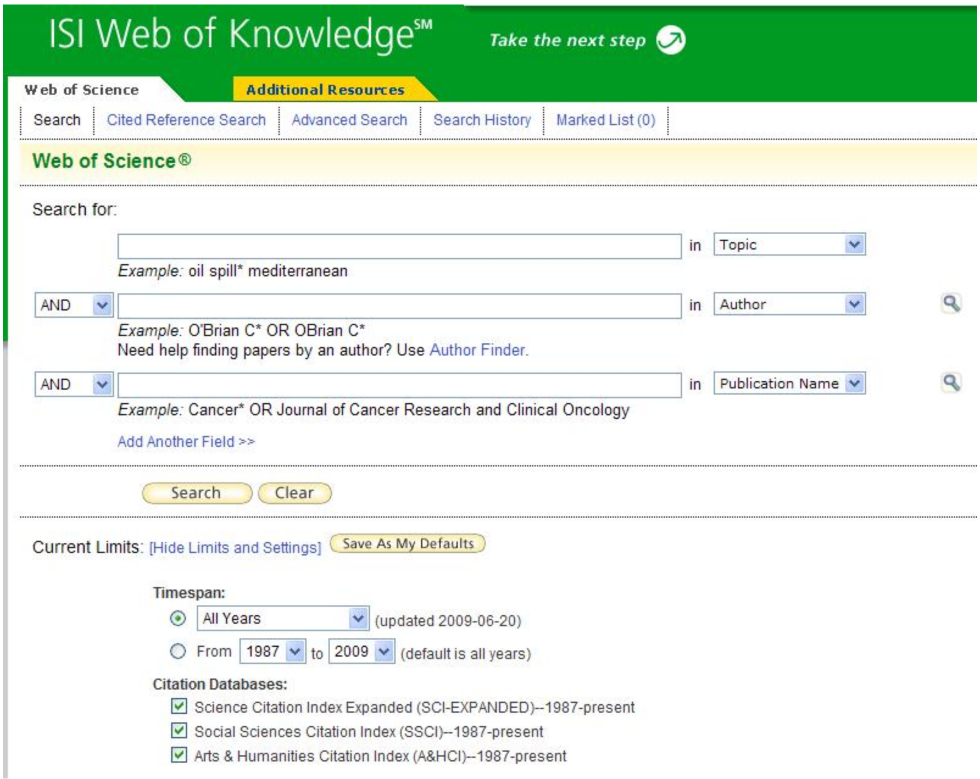
- wcl: WorldCat Local

- europeana: Europeana
- wos: Web of Knowledge

4.4 Testing expressivity

4.4.1 Test 1: Describing basic search in Web of Knowledge

Figure 9 illustrates how the basic search function is described using the DLRM Vocabulary. For better understandability a screenshot of the function being described is provided in Figure 8.



Search for:

in Topic

Example: oil spill* mediterranean

AND in Author

Example: O'Brian C* OR OBrian C*
Need help finding papers by an author? Use Author Finder.

AND in Publication Name

Example: Cancer* OR Journal of Cancer Research and Clinical Oncology

Current Limits:

Timespan:
● All Years (updated 2009-06-20)
○ From 1987 to 2009 (default is all years)

Citation Databases:
☑ Science Citation Index Expanded (SCI-EXPANDED)--1987-present
☑ Social Sciences Citation Index (SSCI)--1987-present
☑ Arts & Humanities Citation Index (A&HCI)--1987-present

Figure 8 Screenshot of basic search in Web of Knowledge (wos:GeneralSearch)

Property	wos:GeneralSearch
actOn	An Information Object (wos:SearchIndex)
functionType	Search@DLRMFunctionTypeVocabulary
influencedBy	An Actor Profile: wos:UserProfile
interactWith	wos:SearchHistory, wos:BrowseSearchResults, wos:Login
return	a ResultSet wos:ResultList (with the format wos:ResultListFormat)
issue	a Query: wos:GeneralSearchQuery
describedBy	An Information Object: http://images.isiknowledge.com/WOK46/help/WOS/h_toc.html
description	Search by metadata fields in selected databases (SCI, A&HI an SCCI) for scientific articles and more
hasFormat	wos:GeneralSearchFormat
hasPart	wos:AddAnotherField, wos:LimitSources, wos:LookUpIndex
hasQuality	a Quality Parameter: wos:GeneralSearchQP1
regulatedBy	a Policy (wos:ISIsAcceptableUsePolicy)
title	Search@en
type	Function@DLRM Type Vocabulary

Function	Description	hasFormat	Type
wos: GeneralSearchFormat	**Function Format:** Input args: wos:GeneralSearchQuery Algorithm: Search in record, included abstract where included (not full text) Interaction mechanism: Text fields (initially 3, but more can be added) + Command button. Implicit AND between fields, but OR/NOT may be used, wos:LimitSourcesFormat		ResourceFormat@ DLRMTypeVocabulary
wos: GeneralSearchQueryFormat	**Query format:** **Components:** Field-based query: - Topic - Title - Author - Group Author - Editor - Publication Name - Year Published - Address - Language - Document type - Funding Agency - Grant Number * Search spec in each field may be combined with AND, OR, NOT * Wildcards (*,?,$) (truncated search) and Boolean opertaors supported in all fields, but the rules vary between fields * Phrase search supported in Topic and Title *Source limits (timespan and database)		ResourceFormat@ DLRMTypeVocabulary
wos: GeneralSearchQuery	A regular query	wos:GeneralSearch QueryFormat	InformationObject @DLRMTypeVocabulary

Figure 9 Description of basic search function in Web of Knowledge (wos:GeneralSearch)

In the topmost table the properties specifically defined for functions are listed first (*actOn* to *issue*), then the properties defined for resources in general. Only properties with values are shown. The properties including their domains and ranges are specified fully in Appendix 3. In the following we will direct special attention to some of the properties:

- Two properties – *type* and *functionType* - are assigned values from the vocabulary encoding schemes derived from DLRM, see [crossref]. The value syntax is <term>@<from-vocabulary>, meaning that <term> is drawn from the vocabulary encoding scheme <from-vocabulary>. The property *type* denotes the main type of resource, which is *Function* for all functions, *Policy* for all policies and so forth. The property *functionType* denotes type of function and is drawn from the *DLRM Function Type Vocabulary*, which is in turn derived from the concept hierarchy under *Function* in DLRM.

- The properties *description* and *describedBy* both represent description of the function. Whereas *description* is assigned a literal value (a string), *describedBy* refers to an information object describing the function

- Two properties – *interactWith* and *hasPart* - connect the described function to other functions. *hasPart* refers to functions that is considered parts of the described function. In the case of Web of Knowledge, the basic search function includes the possibility to add more search fields (`wos:AddAnotherField`), to look up indexes for publications and authors (`wos:LookUpIndex`), and to limit search sources (`wos:LimitSource`). *interactWith* refers to functions that are invoked as a result of, is a prerequisite for, or is influenced by the described function, and vice versa (*interactWith* is symmetric). In the case of Web of Knowledge, the basic search function interacts with the facility for browsing the result list (`wos:BrowseSearchResults`, automatically invoked by the `wos:GeneralSearch`), with the search history function (`wos:SearchHistory`, by `wos:GeneralSearch` adding its query to the search history) and with the login function (`wos:Login`). Note that `wos:Login` interacts with all other functions, indicating the fact that Web of Knowledge can only be used by authorised users. Note also that the functions Advanced Search and Marked List are not considered to interact with `wos:GeneralSearch`, even though they can be invoked from the search screen.

- Some of the properties have values which are objects of other classes, and which must be described separately. Among these are the query issued by `wos:GeneralSearch` (value of the property *issue*: `wos:GeneralSearchQuery`) and the format of the function (value of the property *hasFormat*: `wos:GeneralSearchFormat`). `wos:GeneralSearchQuery` also has its own format: `wos:GeneralSearchQueryFormat`, described in the same table. The arrows in Figure 9 indicate the references.

We now need to investigate how accurately the four characterising elements of a function are described by the tables in Figure 9:

1. Input arguments or initial state: In `wos:GeneralSearch` the input argument is a query inserted by the user, as expressed by the *issue* property. However, for more details about the query, we need to consult `wos:GeneralSearchQuery` itself.

2. Altgorithm: There are no properties of *Function* explicitly intended for information about the action steps or algorithm performed by the function should be described. The value of *actOn* indicates that `wos:GeneralSearch` works on the search index, - a piece of information that might be considered an aspect of the algorithm. For lack of dedicated fields, the remaining information about the algorithm is stored in the *hasFormat* object (`wos:GeneralSearchFormat`), in its *description* field.

3. Output or result: Since `wos:GeneralSearch` has *functionType*=*Search*, and *Search* is a subclass of *Discovery*, the property *return* is applicable. According to the DLRM definition, *return* is meant to represent the collection of search results. Hence the output/result of the function `wos:GeneralSearch` is taken care of.

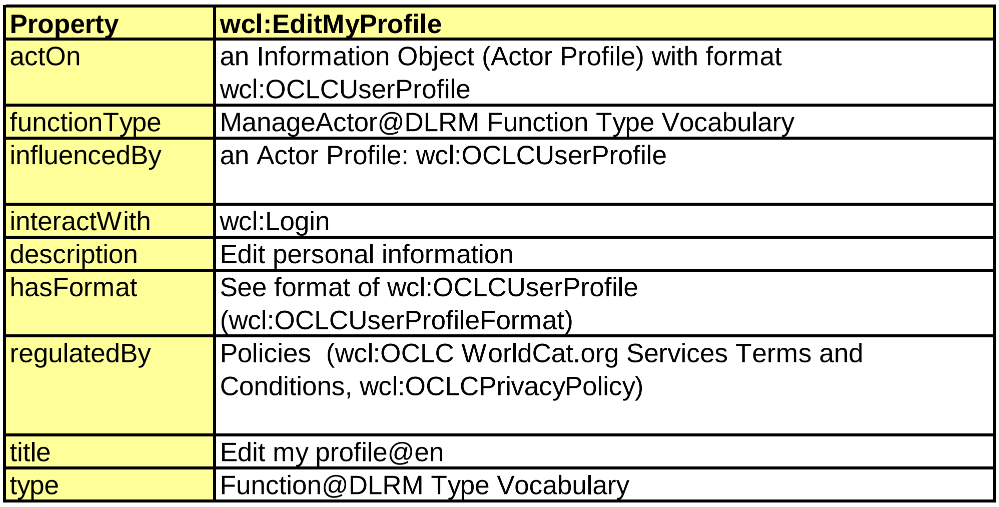
4. Interaction mechanism: There are no explicit properties of Function which indicates that information about the interaction mechanism of the function may be registered. Again, for lack of dedicated fields, the information about the user interaction with `wos:GeneralSearch` is stored in the *hasFormat* object (`wos:GeneralSearchFormat`), in its *description* field.

In the above test, note that I have chosen to view the decomposition of the function into 4 components as the *structure* of the function, and thereby registering this information in the function's *hasFormat* property (see definition of *Resource Format* in Appendix 3). However, the original DLRM specifies only one property for *Resource Format*, namely *expressionOf*, which is meant to reference an ontology expressing a *schema for the format* of the resource type in question (here: *Function*). However, DLRM does not make any instructions, nor provide any ontologies for formats of the various resource types. Such schemas should be specified formally for all resource types before applying DLRM Vocabulary in any serious project. In this test we have only specified the function format informally in the description field, which of course makes it harder to interpret by a computer program.

An alternative way of registering information about the four components of a function could be to use the *hasMetadata* property and define a metadata scheme for functions.

4.4.2 Test 2: Describing editing of personal profile in WorldCat Local

Figure 10 illustrates how the editing of the user profile is described using the DLRM Vocabulary. While not shown in the figure, `wcl:EditMyProfile` has an accompanying format description (`wcl:OCLCUserProfileFormat`), analogous to `wos:GeneralSearch` in Test 1.



Property	wcl:EditMyProfile
actOn	an Information Object (Actor Profile) with format wcl:OCLCUserProfile
functionType	ManageActor@DLRM Function Type Vocabulary
influencedBy	an Actor Profile: wcl:OCLCUserProfile
interactWith	wcl:Login
description	Edit personal information
hasFormat	See format of wcl:OCLCUserProfile (wcl:OCLCUserProfileFormat)
regulatedBy	Policies (wcl:OCLC WorldCat.org Services Terms and Conditions, wcl:OCLCPrivacyPolicy)
title	Edit my profile@en
type	Function@DLRM Type Vocabulary

Figure 10 Description of editing and updating personal profile in WorldCat Local (wcl:EditMyProfile)

We now need to investigate how accurately the four characterising elements of a function are described by the tables in Figure 10:

1. Input arguments or initial state: In `wcl:EditMyProfile` the initial state is the current profile of the current user. The value of *actOn* indicates that `wcl:EditMyProfile` works on the current actor profile (`wcl:OCLCUserProfile`). However, the semantics of *actOn* is not sufficiently

precise to infer that `wcl:OCLCUserProfile` represents the starting point of `wcl:EditMyProfile`.

2. Altgorithm: Same as Test 1

3. Output or result: `wcl:EditMyProfile` updates the user profile, or leaves it unchanged. In any case we want to express that a (possibly updated) user profile is the result of `wcl:EditMyProfile`. However, since `wcl:EditMyProfile` has *functionType=ManageActor*, the property *return* is not applicable. Again we have to resort to the description field of its accompanying resource format (the value of *hasFormat*) to describe this. One could argue that the value of the *actOn* property in this case represents the result, since it indicates the data structure updated by `wcl:EditMyProfile`. But as pointed out in 1 above, the definition of actOn is not precise enough to allow this conclusion.

4. Interaction mechanism: Same as Test 1

4.4.3 Summing up expressivity

The examples in Test 1 and Test 2 show that there are important features of functions (as defined in 4.1) that are not directly expressible in the DLRM Vocabulary in its present state. However, this mainly seems to be a problem connected to the *level of detail and specificity* in the DLRM, not to any major semantic inconsistency between DLRM and the real world as perceived by this author.

- *Relationships between functions:* As already mentioned above, functions can be interconnected using *hasPart* and *interactWith*. (Note that in DLRM, *hasPart* is not specifically constrained to connect resources of the same type, it is entirely possible to express that a function could have other kind of resources, e.g. information objects as parts. However, in this thesis I have constrained the *hasPart* of functions to refer to other functions). And while not specifically declared as symmetric, my interpretation of *interactWith* is that so is the case. At the same time, DLRM states that *interactWith* models the workflow of execution and thereby defines an execution order between the functions. When used symmetrically, however, the property *interactWith* of a function F will refer to functions executed before F as well as functions executed after F , and there is no way of detecting the sequence without further information.

Example: `wos:GeneralSearch` interacts with both `wos:Login` and `wos:BrowseSearchResults` – how do we know the `wos:Login` must be executed first? This issue is also discussed in the analytical evaluation of DLRM, see Chapter 5.

- *Expressing the components of functions:* As pointed out above, DLRM Vocabulary does not prescribe how the structure of resources should be described. It merely states that *Resource Format* is supposed to hold the structure of the resource to which it is attached (through *hasFormat*), but otherwise leaves it almost entirely unspecified. In the rationale of *Resource Format* it says (my clarification in square brackets):

  "For other [than information objects] types of resources, such as users or policies, the schema describes the set of properties or attributes by which the resources are modelled"

  While this supports my choice of *Resource Format* as the container of the four components of *Function*, one could equally well have chosen to use the *hasMetadata* property, for instance. Another possibility could be to model the four components as parts of the function (i.e. values of *hasPart*)

- *The redundancy of actOn:* Since *actOn* merely indicates a resource on which the function operates, there is no way of knowing whether the function manipulates the resource or merely accesses it. This could partly be remedied by using the *functionType* value to make assumptions as to the role of the resource referred to by *actOn*. For example, a function with *functionType=ManageResource* or one of its subtypes could be assumed to manipulate at least one of its *actOn*-s. Likewise, a function with *functionType=Discovery* or one of its subtypes could be assumed not to manipulate any of its *actOn*-s. However, such assumptions would be highly uncertain in the general case. To obtain sufficient specificity regarding this, we need modelling guidelines prescribing which kind of resources can be the value of *actOn*, as well as a way of indicating (in the model) which kind of operation is performed on an *actOn* (access, manage, etc) value.

The main conclusion is that most of the necessary information about functions is possible to express, although in some cases only informally as text. On the other hand, it is clear that DLRM lacks specificity on a number of important points.

To enhance the DLRM Vocabulary as a description vocabulary, it is necessary to define some of the concepts in more detail. Also, to prevent inconsistent modelling practices it is necessary to develop guidelines and instructions for using DLRM Vocabulary.

4.5 Testing discrimination power

In this test we will investigate to what extent differences between functions in the three search systems are reflected by their descriptions.

4.5.1 Test 3: Comparing the basic search functions in Web of Knowledge, WorldCat Local and Europeana

After having studied and used the search functions in all three systems, I observe that the main differences between them from a user point of view are the following:

- *The user interaction mechanism:* Web of Knowledge offers field-based interface, more like a typical "Advanced search" in other systems. Any search word must be tied to a metadata field, but one may use as many fields as need be. WorldCat Local and Europeana offer a single search box in their basic search, although WorldCat Local in addition allows the user to limit sources specifically.

- *The query language:*

  - The query terms in queries to Web of Knowledge are all tagged with field codes (enforced by the user interface, see above). WorldCat Local *allows* field tags in the query string, but if not present all metadata will be searched. Europeana does not support use of field tags in its basic search.

  - In addition, there are differences between the systems regarding use of wildcards. Europeana does not support wildcards, while the other two do, although not exactly the same characters.

  - While all three systems support Boolean operators AND, OR and NOT, the symbols representing them are different. WorldCat Local uses '+', '|','OR' and '-', while the others use 'AND', 'OR', 'NOT'

In Figure 11 and Figure 12 the descriptions of the basic search function in each of the three systems are shown together with their resource format (value of *hasFormat*) and the resource format of their query, respectively.

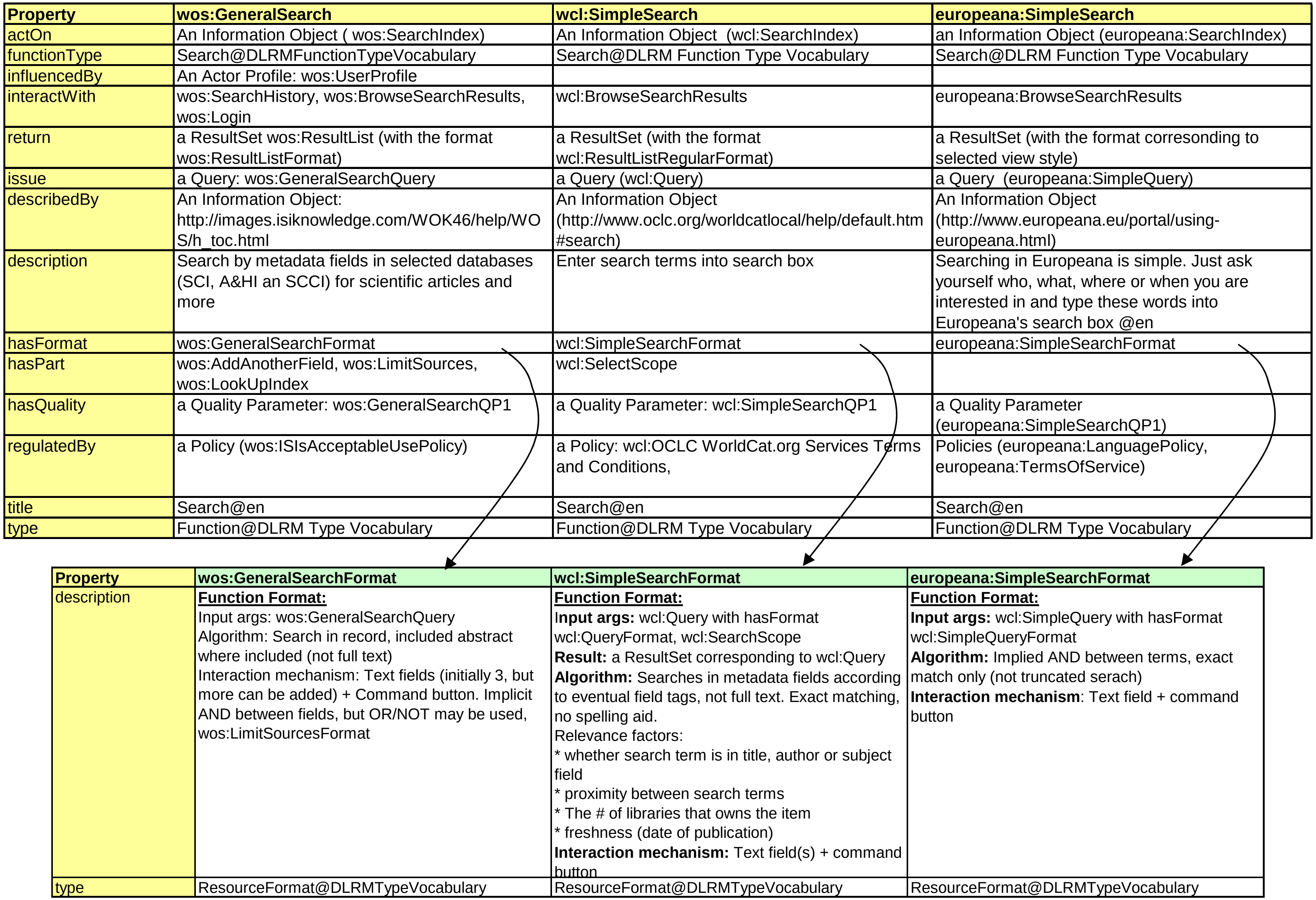
As we can see in Figure 11, the interaction mechanism is described textually in the resource format attached to the functions by the *hasFormat* property. The necessity/motivation for this is discussed at length in previous sections. At this point, suffice it to observe that distinctions founded in properties described in natural language, is not necessarily easy to identify, at least not automatically by computer programs. Even human readers may have problems – the descriptions of the resource format in Figure 11 reflects the challenge in keeping up consistency without support from proper formalisation (and tools). The descriptions, although adhering to the disposition derived from our function "model" are obviously written at separate points in time, and vary in terms of verbosity.

Similar challenges arise when comparing the query languages of the three functions. Query languages are not specifically catered for in DLRM Vocabulary and is therefore stored in the *hasFormat* of the queries, again described in natural language. The encoding of the query format in Figure 12 indicates some effort on the part of the encoder to keep the descriptions stringent and comparable, but without proper formalisation in representation as well as practice, this stringency is hard to keep up over time.

4.5.2 Test 4: Comparing quality parameters

The DLRM Vocabulary allows quality parameters to be connected to any resource through the *hasQuality* property. Figure 13 illustrates how this can be used, and tests whether the quality assignments are comparable. In this example, a scale 1-6 is used to assign a subjective usability score to each of the functions in Web of Knowledge and Europeana, whereas a scale 1-4 is used to assign usability score to WorldCat Local. In DLRM terms, quality parameters with *qualityParameterType=Usability* (selected from DLRM Quality Parameter Type Vocabulary) are connected to each of the functions. Each quality parameter is *evaluatedBy* a measure representing the procedure or principle by which the quality parameter is measured, and *measuredBy* a numeric value representing the score according to the measure. All measures are assigned two measure types, - *measureType=SubjectiveMeasure* and *measureType=QualitativeMeasure*. However, for describing the actual measuring procedure or formula we again must resort to the description field. Hence, even though it is easy to verify that all three quality parameters are assessed using the same measure type, and by the same actor, the specific measuring procedure is not equally discernable. Not until we are able to distinguish between the measures are we able to conclude that the usability score of "4" to

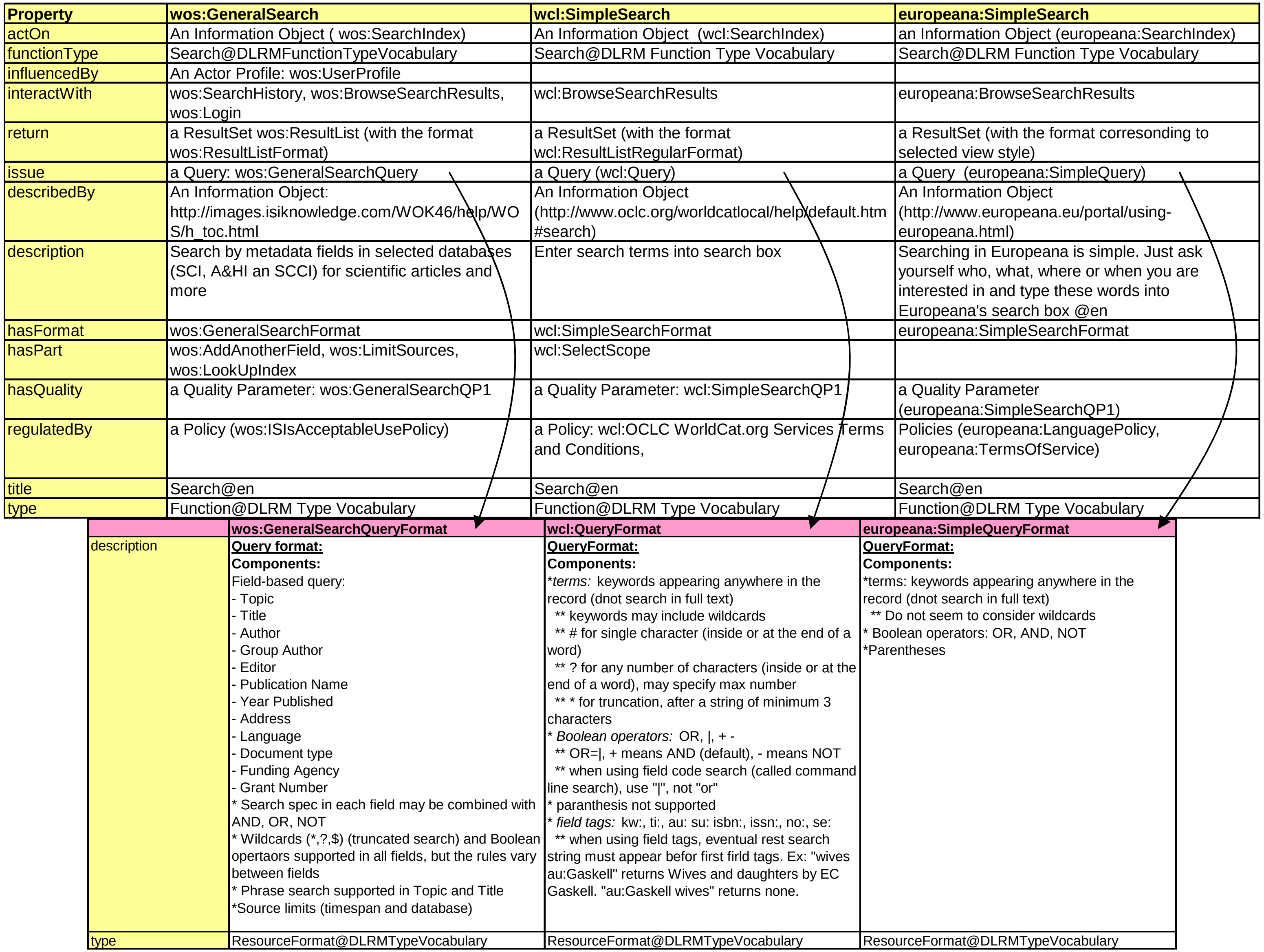
Europeana and WorldCat Local actually represent very different assessments of their usability. And for that, the ability of interpreting natural language is necessary.



Property	wos:GeneralSearch	wcl:SimpleSearch	europeana:SimpleSearch
actOn	An Information Object (wos:SearchIndex)	An Information Object (wcl:SearchIndex)	an Information Object (europeana:SearchIndex)
functionType	Search@DLRMFunctionTypeVocabulary	Search@DLRM Function Type Vocabulary	Search@DLRM Function Type Vocabulary
influencedBy	An Actor Profile: wos:UserProfile		
interactWith	wos:SearchHistory, wos:BrowseSearchResults, wos:Login	wcl:BrowseSearchResults	europeana:BrowseSearchResults
return	a ResultSet wos:ResultList (with the format wos:ResultListFormat)	a ResultSet (with the format wcl:ResultListRegularFormat)	a ResultSet (with the format corresonding to selected view style)
issue	a Query: wos:GeneralSearchQuery	a Query (wcl:Query)	a Query (europeana:SimpleQuery)
describedBy	An Information Object: http://images.isiknowledge.com/WOK46/help/WOS/h_toc.html	An Information Object (http://www.oclc.org/worldcatlocal/help/default.htm#search)	An Information Object (http://www.europeana.eu/portal/using-europeana.html)
description	Search by metadata fields in selected databases (SCI, A&HI an SCCI) for scientific articles and more	Enter search terms into search box	Searching in Europeana is simple. Just ask yourself who, what, where or when you are interested in and type these words into Europeana's search box @en
hasFormat	wos:GeneralSearchFormat	wcl:SimpleSearchFormat	europeana:SimpleSearchFormat
hasPart	wos:AddAnotherField, wos:LimitSources, wos:LookUpIndex	wcl:SelectScope	
hasQuality	a Quality Parameter: wos:GeneralSearchQP1	a Quality Parameter: wcl:SimpleSearchQP1	a Quality Parameter (europeana:SimpleSearchQP1)
regulatedBy	a Policy (wos:ISIsAcceptableUsePolicy)	a Policy: wcl:OCLC WorldCat.org Services Terms and Conditions,	Policies (europeana:LanguagePolicy, europeana:TermsOfService)
title	Search@en	Search@en	Search@en
type	Function@DLRM Type Vocabulary	Function@DLRM Type Vocabulary	Function@DLRM Type Vocabulary

Property	wos:GeneralSearchFormat	wcl:SimpleSearchFormat	europeana:SimpleSearchFormat
description	<u>**Function Format:**</u> Input args: wos:GeneralSearchQuery Algorithm: Search in record, included abstract where included (not full text) Interaction mechanism: Text fields (initially 3, but more can be added) + Command button. Implicit AND between fields, but OR/NOT may be used, wos:LimitSourcesFormat	<u>**Function Format:**</u> **Input args:** wcl:Query with hasFormat wcl:QueryFormat, wcl:SearchScope **Result:** a ResultSet corresponding to wcl:Query **Algorithm:** Searches in metadata fields according to eventual field tags, not full text. Exact matching, no spelling aid. Relevance factors: * whether search term is in title, author or subject field * proximity between search terms * The # of libraries that owns the item * freshness (date of publication) **Interaction mechanism:** Text field(s) + command button	<u>**Function Format:**</u> **Input args:** wcl:SimpleQuery with hasFormat wcl:SimpleQueryFormat **Algorithm:** Implied AND between terms, exact match only (not truncated serach) **Interaction mechanism**: Text field + command button
type	ResourceFormat@DLRMTypeVocabulary	ResourceFormat@DLRMTypeVocabulary	ResourceFormat@DLRMTypeVocabulary

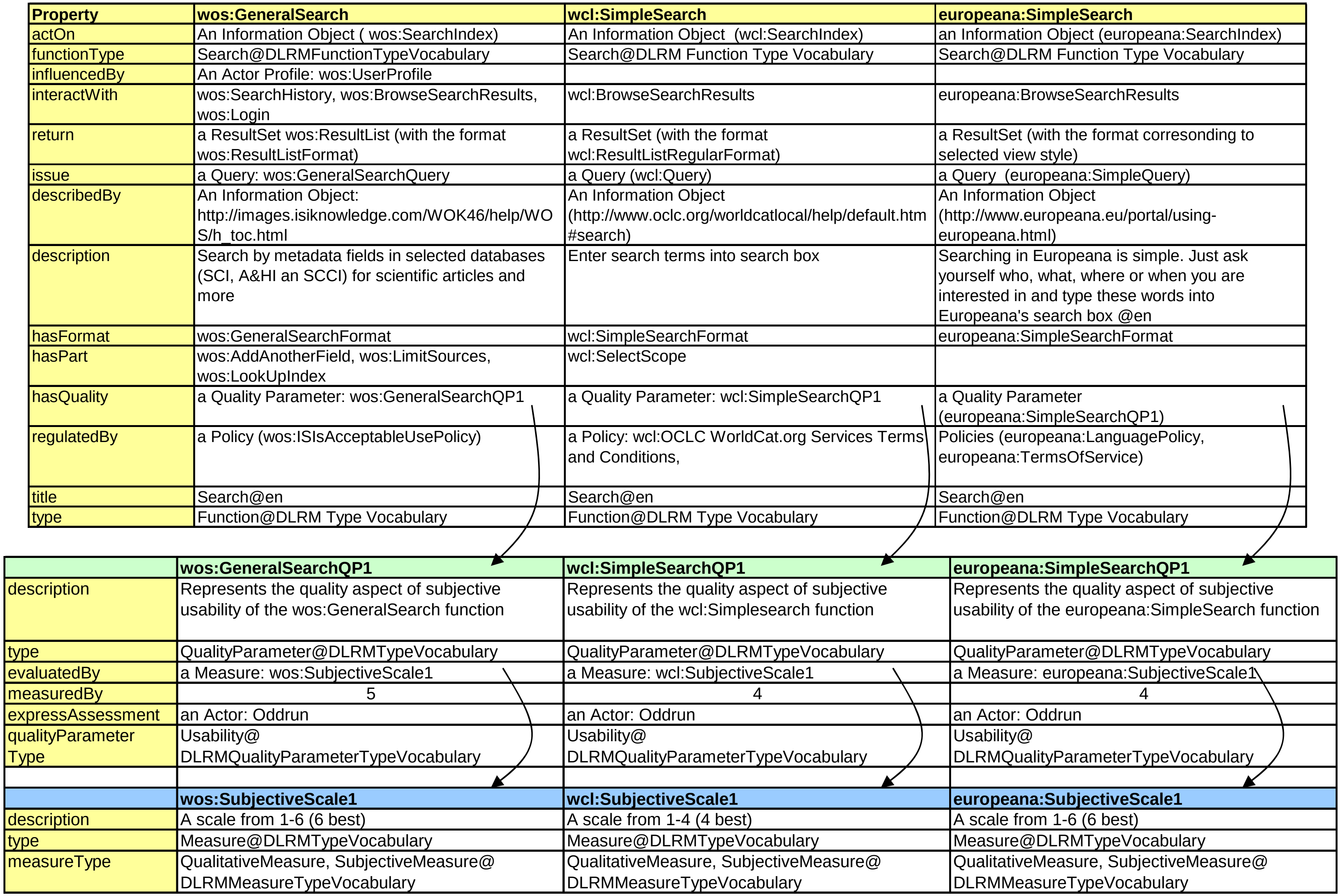
Figure 11 Basic search functions in Web of Knowledge, Worldcat Local and Europeana, with details about their resource format



Property	wos:GeneralSearch	wcl:SimpleSearch	europeana:SimpleSearch
actOn	An Information Object (wos:SearchIndex)	An Information Object (wcl:SearchIndex)	an Information Object (europeana:SearchIndex)
functionType	Search@DLRMFunctionTypeVocabulary	Search@DLRM Function Type Vocabulary	Search@DLRM Function Type Vocabulary
influencedBy	An Actor Profile: wos:UserProfile		
interactWith	wos:SearchHistory, wos:BrowseSearchResults, wos:Login	wcl:BrowseSearchResults	europeana:BrowseSearchResults
return	a ResultSet wos:ResultList (with the format wos:ResultListFormat)	a ResultSet (with the format wcl:ResultListRegularFormat)	a ResultSet (with the format corresonding to selected view style)
issue	a Query: wos:GeneralSearchQuery	a Query (wcl:Query)	a Query (europeana:SimpleQuery)
describedBy	An Information Object: http://images.isiknowledge.com/WOK46/help/WOS/h_toc.html	An Information Object (http://www.oclc.org/worldcatlocal/help/default.htm#search)	An Information Object (http://www.europeana.eu/portal/using-europeana.html)
description	Search by metadata fields in selected databases (SCI, A&HI an SCCI) for scientific articles and more	Enter search terms into search box	Searching in Europeana is simple. Just ask yourself who, what, where or when you are interested in and type these words into Europeana's search box @en
hasFormat	wos:GeneralSearchFormat	wcl:SimpleSearchFormat	europeana:SimpleSearchFormat
hasPart	wos:AddAnotherField, wos:LimitSources, wos:LookUpIndex	wcl:SelectScope	
hasQuality	a Quality Parameter: wos:GeneralSearchQP1	a Quality Parameter: wcl:SimpleSearchQP1	a Quality Parameter (europeana:SimpleSearchQP1)
regulatedBy	a Policy (wos:ISIsAcceptableUsePolicy)	a Policy: wcl:OCLC WorldCat.org Services Terms and Conditions,	Policies (europeana:LanguagePolicy, europeana:TermsOfService)
title	Search@en	Search@en	Search@en
type	Function@DLRM Type Vocabulary	Function@DLRM Type Vocabulary	Function@DLRM Type Vocabulary

	wos:GeneralSearchQueryFormat	wcl:QueryFormat	europeana:SimpleQueryFormat
description	**Query format:** **Components:** Field-based query: - Topic - Title - Author - Group Author - Editor - Publication Name - Year Published - Address - Language - Document type - Funding Agency - Grant Number * Search spec in each field may be combined with AND, OR, NOT * Wildcards (*,?,$) (truncated search) and Boolean opertaors supported in all fields, but the rules vary between fields * Phrase search supported in Topic and Title *Source limits (timespan and database)	**QueryFormat:** **Components:** *_terms:_ keywords appearing anywhere in the record (dnot search in full text) ** keywords may include wildcards ** # for single character (inside or at the end of a word) ** ? for any number of characters (inside or at the end of a word), may specify max number ** * for truncation, after a string of minimum 3 characters * _Boolean operators:_ OR, |, + - ** OR=|, + means AND (default), - means NOT ** when using field code search (called command line search), use "|", not "or" * paranthesis not supported * _field tags:_ kw:, ti:, au: su: isbn:, issn:, no:, se: ** when using field tags, eventual rest search string must appear befor first firld tags. Ex: "wives au:Gaskell" returns Wives and daughters by EC Gaskell. "au:Gaskell wives" returns none.	**QueryFormat:** **Components:** *terms: keywords appearing anywhere in the record (dnot search in full text) ** Do not seem to consider wildcards * Boolean operators: OR, AND, NOT *Parentheses
type	ResourceFormat@DLRMTypeVocabulary	ResourceFormat@DLRMTypeVocabulary	ResourceFormat@DLRMTypeVocabulary

Figure 12 Basic search functions in Web of Knowledge, Worldcat Local and Europeana, with details about their query language



Property	wos:GeneralSearch	wcl:SimpleSearch	europeana:SimpleSearch
actOn	An Information Object (wos:SearchIndex)	An Information Object (wcl:SearchIndex)	an Information Object (europeana:SearchIndex)
functionType	Search@DLRMFunctionTypeVocabulary	Search@DLRM Function Type Vocabulary	Search@DLRM Function Type Vocabulary
influencedBy	An Actor Profile: wos:UserProfile		
interactWith	wos:SearchHistory, wos:BrowseSearchResults, wos:Login	wcl:BrowseSearchResults	europeana:BrowseSearchResults
return	a ResultSet wos:ResultList (with the format wos:ResultListFormat)	a ResultSet (with the format wcl:ResultListRegularFormat)	a ResultSet (with the format corresonding to selected view style)
issue	a Query: wos:GeneralSearchQuery	a Query (wcl:Query)	a Query (europeana:SimpleQuery)
describedBy	An Information Object: http://images.isiknowledge.com/WOK46/help/WOS/h_toc.html	An Information Object (http://www.oclc.org/worldcatlocal/help/default.htm#search)	An Information Object (http://www.europeana.eu/portal/using-europeana.html)
description	Search by metadata fields in selected databases (SCI, A&HI an SCCI) for scientific articles and more	Enter search terms into search box	Searching in Europeana is simple. Just ask yourself who, what, where or when you are interested in and type these words into Europeana's search box @en
hasFormat	wos:GeneralSearchFormat	wcl:SimpleSearchFormat	europeana:SimpleSearchFormat
hasPart	wos:AddAnotherField, wos:LimitSources, wos:LookUpIndex	wcl:SelectScope	
hasQuality	a Quality Parameter: wos:GeneralSearchQP1	a Quality Parameter: wcl:SimpleSearchQP1	a Quality Parameter (europeana:SimpleSearchQP1)
regulatedBy	a Policy (wos:ISIsAcceptableUsePolicy)	a Policy: wcl:OCLC WorldCat.org Services Terms and Conditions,	Policies (europeana:LanguagePolicy, europeana:TermsOfService)
title	Search@en	Search@en	Search@en
type	Function@DLRM Type Vocabulary	Function@DLRM Type Vocabulary	Function@DLRM Type Vocabulary

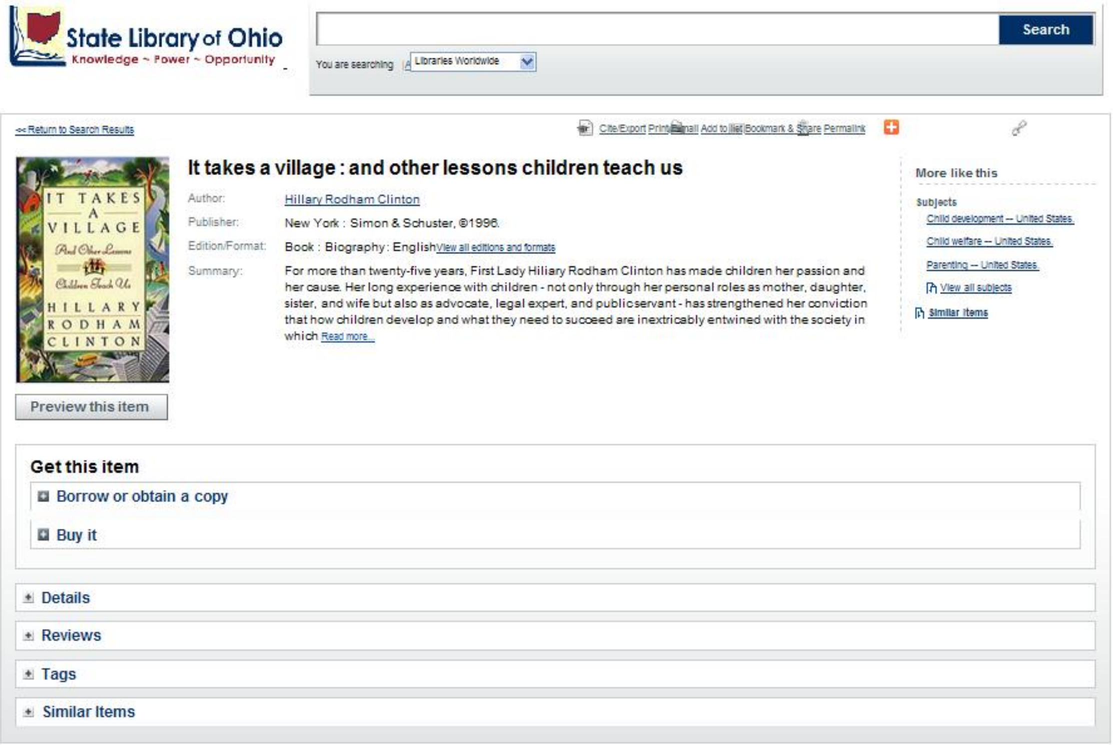
	wos:GeneralSearchQP1	wcl:SimpleSearchQP1	europeana:SimpleSearchQP1
description	Represents the quality aspect of subjective usability of the wos:GeneralSearch function	Represents the quality aspect of subjective usability of the wcl:Simplesearch function	Represents the quality aspect of subjective usability of the europeana:SimpleSearch function
type	QualityParameter@DLRMTypeVocabulary	QualityParameter@DLRMTypeVocabulary	QualityParameter@DLRMTypeVocabulary
evaluatedBy	a Measure: wos:SubjectiveScale1	a Measure: wcl:SubjectiveScale1	a Measure: europeana:SubjectiveScale1
measuredBy	5	4	4
expressAssessment	an Actor: Oddrun	an Actor: Oddrun	an Actor: Oddrun
qualityParameter Type	Usability@ DLRMQualityParameterTypeVocabulary	Usability@ DLRMQualityParameterTypeVocabulary	Usability@ DLRMQualityParameterTypeVocabulary

	wos:SubjectiveScale1	wcl:SubjectiveScale1	europeana:SubjectiveScale1
description	A scale from 1-6 (6 best)	A scale from 1-4 (4 best)	A scale from 1-6 (6 best)
type	Measure@DLRMTypeVocabulary	Measure@DLRMTypeVocabulary	Measure@DLRMTypeVocabulary
measureType	QualitativeMeasure, SubjectiveMeasure@ DLRMMeasureTypeVocabulary	QualitativeMeasure, SubjectiveMeasure@ DLRMMeasureTypeVocabulary	QualitativeMeasure, SubjectiveMeasure@ DLRMMeasureTypeVocabulary

Figure 13 Basic search functions in Web of Knowledge, Worldcat Local and Europeana, with details about a quality parameter assigned to them

4.5.3 Test 5: Comparing single record screens

The function for displaying a selected single record is typically invoked by clicking at a title in a search result list. Examples of the single record screen for each of the three systems are shown in the next three figures.


State Library of Ohio
Knowledge ~ Power ~ Opportunity
Search
You are searching
Libraries Worldwide
<< Return to Search Results
Cite/Export Print Email Add to list Bookmark & Share Permalink
Preview this item
More like this

It takes a village : and other lessons children teach us

Author: Hillary Rodham Clinton

Publisher: New York : Simon & Schuster, ©1996.

Edition/Format: Book : Biography : English View all editions and formats

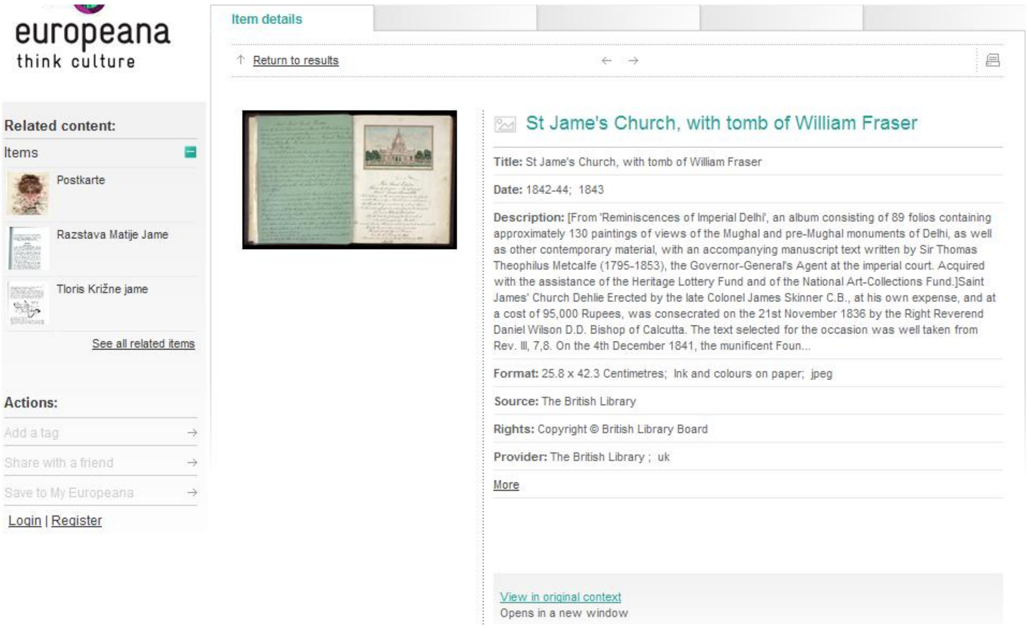
Summary: For more than twenty-five years, First Lady Hiliary Rodham Clinton has made children her passion and her cause. Her long experience with children - not only through her personal roles as mother, daughter, sister, and wife but also as advocate, legal expert, and public servant - has strengthened her conviction that how children develop and what they need to succeed are inextricably entwined with the society in which Read more...

Subjects

- Child development -- United States.
- Child welfare -- United States.
- Parenting -- United States.
- View all subjects

Similar Items

Figure 14 Detailed record screen from WorldCat Local at State Library of Ohio (wcl:ViewSingleScreen)



St Jame's Church, with tomb of William Fraser

Title: St Jame's Church, with tomb of William Fraser

Date: 1842-44; 1843

Description: [From 'Reminiscences of Imperial Delhi', an album consisting of 89 folios containing approximately 130 paintings of views of the Mughal and pre-Mughal monuments of Delhi, as well as other contemporary material, with an accompanying manuscript text written by Sir Thomas Theophilus Metcalfe (1795-1853), the Governor-General's Agent at the imperial court. Acquired with the assistance of the Heritage Lottery Fund and of the National Art-Collections Fund.]Saint James' Church Dehlie Erected by the late Colonel James Skinner C.B., at his own expense, and at a cost of 95,000 Rupees, was consecrated on the 21st November 1836 by the Right Reverend Daniel Wilson D.D. Bishop of Calcutta. The text selected for the occasion was well taken from Rev. III, 7,8. On the 4th December 1841, the munificent Foun...

Format: 25.8 x 42.3 Centimetres; Ink and colours on paper; jpeg

Source: The British Library

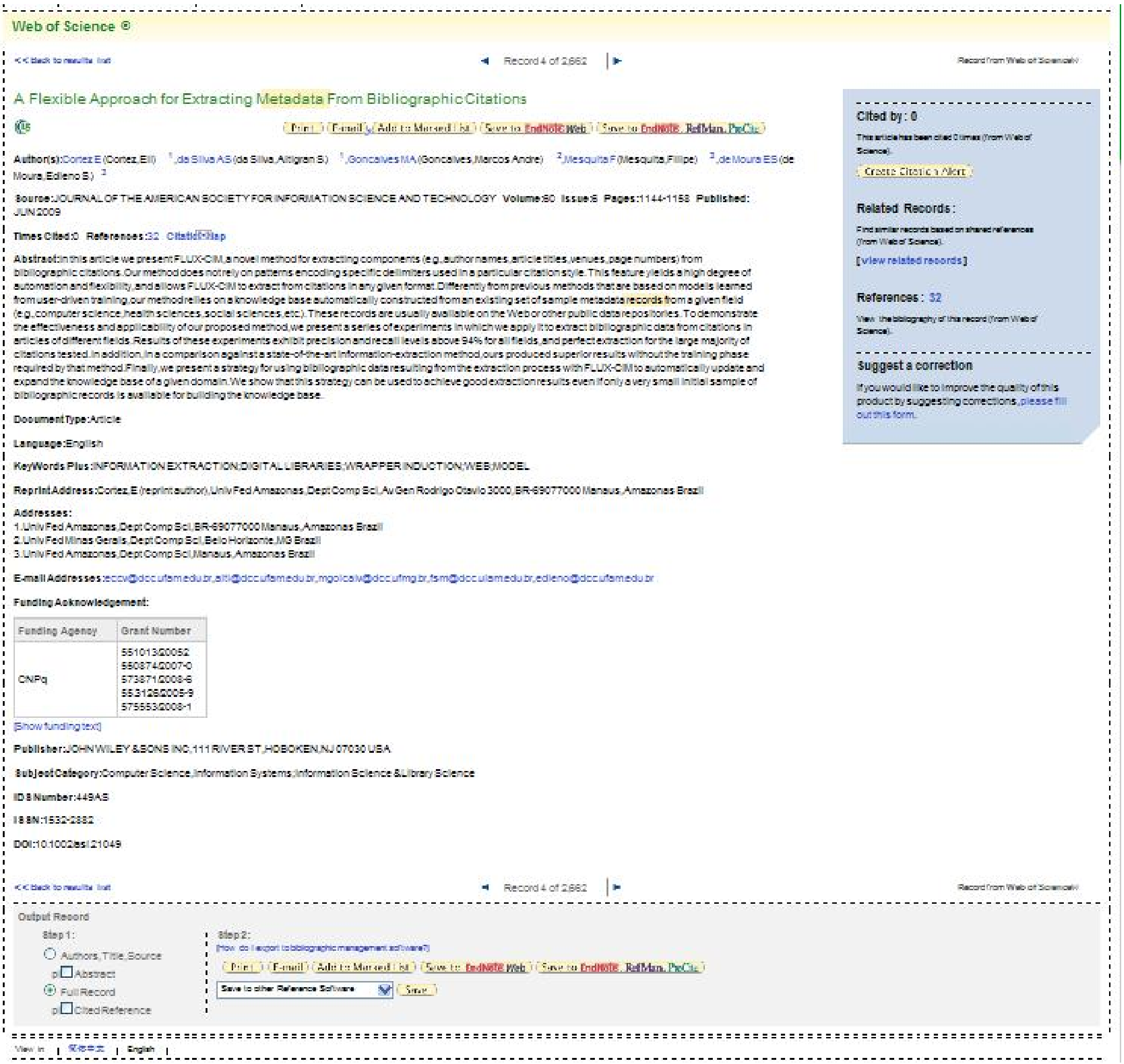
Rights: Copyright © British Library Board

Provider: The British Library ; uk

More

View in original context
Opens in a new window

Figure 15 View single record from Europeana (europeana:ViewSingleResource)



Web of Science ®

<< back to results list | Record 4 of 2,662 | Record from Web of Science®

A Flexible Approach for Extracting Metadata From Bibliographic Citations

Author(s):Cortez E (Cortez,Eli) [1],da Silva AS (da Silva,Altigran S.) [1],Goncalves MA (Goncalves,Marcos Andre) [2],Mesquita F (Mesquita,Fillipe) [2],de Moura ES (de Moura,Edleno S.) [3]

Source:JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE AND TECHNOLOGY Volume:60 Issue:6 Pages:1144-1158 Published: JUN 2009

Times Cited:0 References:32 Citation Map

Abstract:In this article we present FLUX-CIM,a novel method for extracting components (e.g.,author names,article titles,venues,page numbers) from bibliographic citations.Our method does not rely on patterns encoding specific delimiters used in a particular citation style. This feature yields a high degree of automation and flexibility,and allows FLUX-CIM to extract from citations in any given format.Differently from previous methods that are based on models learned from user-driven training,our method relies on a knowledge base automatically constructed from an existing set of sample metadata records from a given field (e.g.,computer science,health sciences,social sciences,etc.). These records are usually available on the Web or other public data repositories. To demonstrate the effectiveness and applicability of our proposed method,we present a series of experiments in which we apply it to extract bibliographic data from citations in articles of different fields.Results of these experiments exhibit precision and recall levels above 94% for all fields,and perfect extraction for the large majority of citations tested.In addition,in a comparison against a state-of-the-art information-extraction method,ours produced superior results without the training phase required by that method.Finally,we present a strategy for using bibliographic data resulting from the extraction process with FLUX-CIM to automatically update and expand the knowledge base of a given domain.We show that this strategy can be used to achieve good extraction results even if only a very small initial sample of bibliographic records is available for building the knowledge base.

Document Type:Article

Language:English

KeyWords Plus:INFORMATION EXTRACTION;DIGITAL LIBRARIES;WRAPPER INDUCTION;WEB;MODEL

Reprint Address:Cortez,E (reprint author),Univ Fed Amazonas,Dept Comp Sci,Av Gen Rodrigo Otavio 3000,BR-69077000 Manaus,Amazonas Brazil

Addresses:
1.Univ Fed Amazonas,Dept Comp Sci,BR-69077000 Manaus,Amazonas Brazil
2.Univ Fed Minas Gerais,Dept Comp Sci,Belo Horizonte,MG Brazil
3.Univ Fed Amazonas,Dept Comp Sci,Manaus,Amazonas Brazil

E-mail Addresses:ecov@dcc.ufam.edu.br,alti@dcc.ufam.edu.br,mgoncalv@dcc.ufmg.br,fsm@dcc.ufam.edu.br,edleno@dcc.ufam.edu.br

Funding Acknowledgement:

Funding Agency	Grant Number
CNPq	551013/2005-2 550874/2007-0 573871/2008-6 553126/2005-9 575553/2008-1

[Show funding text]

Publisher:JOHN WILEY & SONS INC,111 RIVER ST,HOBOKEN,NJ 07030 USA

Subject Category:Computer Science,Information Systems;Information Science & Library Science

IDS Number:449AS

ISSN:1532-2882

DOI:10.1002/asi.21049

<< back to results list | Record 4 of 2,662 | Record from Web of Science®

Cited by : 0

This article has been cited 0 times (from Web of Science).

Related Records :

Find similar records based on shared references (from Web of Science).

[view related records]

References : 32

View the bibliography of this record (from Web of Science).

Suggest a correction

If you would like to improve the quality of this product by suggesting corrections, please fill out this form.

Output Record

Step 1:
○ Authors, Title, Source
□ Abstract
● Full Record
□ Cited Reference

Step 2:
[How do I export to bibliographic management software?]

Print
E-mail
Add to Marked List
Save to EndNote Web
Save to EndNote, RefMan, ProCite
Save to other Reference Software
Save
View in | 简体中文 | English |

Figure 16 Full record from Web of knowledge (wos:ViewSingleRecord)

The main difference between the single record function across systems is the amount of functionality available related to the resource shown. Europeana offers a modest amount of things to do based on the single record, there are some sharing-with-others facilities, it is possible to save the record for later and there is also offered an opportunity to view related material. Web of Knowledge and WorldCat Local both offers rich functionality based on the single record, one marked difference being the multitude of WorldCat Local functions related to getting hold of the resource.

Property	wos:ViewSingleRecord	wcl:ViewSingleRecord	europeana: ViewSingleResource
actOn	An Information Object (selected record from ResultSet)	an Information Object selected from the ResultSet (with the format wcl:SingleRecordFormat)	an Information Object (selected record in ResultSet)
functionType	AcessResource@DLRM Function Type Vocabulary	AcessResource@DLRM Function Type Vocabulary	AccessResource@DLRM Function Type Vocabulary
influencedBy	An Actor Profile: wos:UserProfile		
interactWith	wos:Login, wos:BrowseSearchResults	wcl:BrowseSearchResults	europeana:BrowseSearchResults (all parts), europeana:BrowsePreselected, europeana:TimeLineNavigator, europeana:BrowseTime,
describedBy	An Information Object: http://images.isiknowledge.com/WOK46/help/WOS/h_toc.html	An Information Object (http://www.oclc.org/worldcatlocal/help/default.htm)	An Information Object (http://www.europeana.eu/portal/using-europeana.html)
description	View information about a single record	View information about a single record from Result Set	View all information Europeana has about the selected Information Object
hasFormat	wos:ViewSingleRecordFormat	a Resource Format for functions:wcl:ViewSingleRecordFormat	A Resource Format for Information Objects: europeana:FullRecordFormat
hasPart	wos:GetFullText, wos:ContextSensitiveGetItem, wos:OutputMarkedRecord, wos:CreateCitationMap, wos:CreateCitationAlert, wos:ViewRelatedRecords, wos:ViewReferences, wos:SuggestCorrection	wcl:CiteRecord, wcl:PrintRecord, wcl:EmailRecord, wcl:AddRecordToList, wcl:BookmarkAndShare, wcl:Permalink, wcl:MoreLikeThisSubjects, wcl:MoreLikeThisSimilarItems, wcl:PreviewItem, wcl:GetItem, wcl:MoreByContributors, wcl:AboutContributors, wcl:WriteReview, wcl:AddTags, wcl:OtherArticlesInPublication, wcl:ViewAllEditionsAndFormats	europeana:ViewOriginal, europeana:ViewRelatedContent, europeana:AddTag, europeana:ShareWithFriend, europeana:SaveRecord
hasQuality		a Quality Parameter: wcl:ViewSingleRecordQP1	
regulatedBy	a Policy (wos:ISIsAcceptableUsePolicy)	a Policy (wcl:OCLC WorldCat.org Services Terms and Conditions)	europeana:LanguagePolicy, europeana:TermsOfService
title	Full record@en	Detailed record screen@en	View single record@en
type	Function@DLRM Type Vocabulary	Function@DLRM Type Vocabulary	Function@DLRM Type Vocabulary

Figure 17 The functions for displaying single records in Web of knowledge, WorldCat Local and Europeana

Considering that the subfunctions of a function are recorded as values of the hasPart property, the distinctions between the single record view across systems should be detectable by comparing the hasPart property. From Figure 17 above it evident that Worldcat Local offers much more functionality than the other two, and that Europeana offers much less than the other two. To investigate further into the differences, the functionType of each of the subfunctions are retrieved and listed in the table below, see Appendix 3 for definition of *functionType* terms. Note that *Acquire*, *Browse*, *Search*, and *Visualise* are all subclasses of *AccessResource*. In the table this is visualised by the *AccessResource* row containing the sums of that of its subclasses, in addition to the number of functions having been classified as *AccessResource* directly.

functionType	**Web of Knowledge**	**WorldCat Local**	**Europeana**
AccessResource	6	12	3
Acquire	3	6	1
Browse	1	2	1
Collaborate	1	2	1
Manage-InformationObject	2	2	1
Search	1	4	0
Visualise	1	0	0
Total number of functions	**8**	**16**	**5**

Table 1 Number and types of functions offered in the single record view in Web of knowledge, WorldCat Local and Europeana.

From this we can see that WorldCat Local has markedly more ways of acquiring an item than the other two, which is natural, seeing that WorldCat Local is a library catalogue. However, it also offers more ways of finding new information on the basis of a single record, reflected by 2 *Browse* and 4 *Search* functions.

4.5.4 Summing up discrimination power

Discrimination power is of course dependent on expressivity, which is evident from the tests above. In the cases where the distinctions across systems are connected to properties for which expressivity is a problem, the discrimination power is naturally low. Hence, since neither the function “model” nor the query “model” – as described using *Resource Format* - are in any way formalised, we have had to use plain language to describe the function format (in terms of input arguments, algorithm, output/result and interaction mechanism) and query format (in terms of allowable query components).

On the other hand, in cases where the major distinctions between functions are based on number and types of their subfunctions, the descriptions based on the current DLRM Vocabulary are quite informative, as indicated by Test 5 in 4.5.3 above. For anyone who is familiar with the DLRM Vocabulary, Table 1 gives a good overview of the differences between Web of Knowledge, WorldCat Local and Europeana concerning functionality available in the single record view.

5 Analytical evaluation of the DLRM Vocabulary

The word “model” is used about many things, but in general it denotes an *abstraction* of something in the real world, like a company (an enterprise architecture model), a house (architecture drawing) or an information system (information model). Models are created using modelling languages, which provides a conceptual basis (a collection of modelling constructs) and (most often) a modelling notation. The latter may take many forms; graphical, textual or symbolic. Example: BPMN (OMG) is a business process modelling language, and its conceptual basis is the various constructs offered to describe processes (activity, sequence flow, data object etc), whereas its notation is the collection of graphical symbols with which to depict the various constructs.

A reference model is a model which represents not one particular instance, but a class of things that in some sense are perceived to be “similar” or “related”. DLRM, being a reference model for digital libraries, denotes a “generic” digital library, meant to contain most of the entities that are needed to describe individual digital libraries.

In that sense a reference model might be viewed as a modelling language, a language by which objects in the domain of discourse may be described. We therefore argue that quality principles defined for modelling languages may be used to evaluate reference models. This implies that the DLRM Vocabulary, the description profile derived from the original DLRM may be evaluated as a digital library modelling language.

5.1 Quality of modelling languages

In (J. Krogstie, 2008) and (J. Krogstie, 2003) a generic framework for evaluating the quality of conceptual modelling languages is presented. In (J. Krogstie, 2003) it is used to evaluate the quality of UML (OMG), and in (Wahl & Sindre, 2006) the framework is used to perform an analytical evaluation of the business process modelling language BPMN (OMG). The quality framework for modelling languages is an extension of a more general framework for quality of conceptual models, SEQUAL (J. Krogstie, 2003, 2008).

SEQUAL operates with seven types of quality of conceptual models:

- Physical quality: has to do with the persistence and availability of the model, that is the opportunity for the audience to study reliable specimens of the model and make sense of it;

- Empirical quality has to do with readability of the model, generally affected by layout and presentation;

- Syntactic quality is concerned with the model being expressed correctly according to the syntax and vocabulary of the modelling language used;

- Semantic quality has to do with the correspondence between the model and the domain that is modelled. Both *validity* (all statements in the model are valid according to the domain) and *completeness* (all relevant statements of the domain are expressed in the model) are important *Perceived semantic quality* is the semantic quality "as an actor sees it" and denotes the correspondence between interpretation of a model by an actor and the actor's current knowledge of the domain;

- Pragmatic quality regards the intended audience's comprehension of the model, how well they are able to understand it and thereby make use of it;

- Social quality has to do with agreement among different actors. Actors may agree or disagree on how the model should be interpreted or on domain knowledge as such;

- Organisational quality concerns how models contribute towards achieving organisational goals.

The quality framework for modelling *languages* comprise six criteria, of which the following five are relevant in our case. Each criterion contribute to one or more types of quality listed above. The criteria are:

1. *Domain appropriateness* relates the modelling language to the domain. The conceptual basis of a modelling language should be powerful enough to represent any fact of the domain. On the other hand the language should not make it possible to express facts not in the domain. Contributes towards semantic quality

2. *Participant language knowledge appropriateness*: This criterion is about how well the participants' knowledge of modelling languages fits the actual modelling language, and how well the conceptual basis for the language corresponds to the participants' perception of reality (the domain of discourse). Ideally they should know all the constructs of the language, and understand

their use. Contributes towards semantic or pragmatic quality (depending on the participant's role);

3. *Knowledge externalisability appropriateness* (also called modeller's knowledge appropriateness): Relates the language to the explicit (externalizable) knowledge of the person doing the actual modelling. The aim is that the modeller is able to express all his knowledge (relevant to the domain in question) by means of the modelling language. Contributes towards semantic quality.

4. *Comprehensibility appropriateness:* Relates the language to the interpretation of the social actors (the audience) involved in the modelling effort. Ideally the audience should be able to understand any statement possible to make in the language in question. Contributes towards empirical and pragmatic quality

5. *Technical actor interpretation appropriateness (tool appropriateness)*: Concerns how easy the language lends itself to tool interpretation, especially automatic reasoning. This corresponds to the degree of formality of the language. Provided that both syntax and semantics are formally defined, and the tool is able to process the statements efficiently, it is possible to perform automatic reasoning based on the explicit statements in the model. Contributes towards syntactic, semantic and pragmatic quality.

5.2 The quality of the DLRM Vocabulary

In the following the framework will be applied to evaluate the DLRM Vocabulary.

5.2.1 Domain appropriateness

Our domain of discourse is *digital libraries* in a very broad and comprehensive sense. Hence, domain appropriateness in this case is about *being able to express* things and phenomena about digital libraries. DLRM Vocabulary is maximally appropriate for the DL domain if it is able to express *everything that is relevant* about digital libraries and *nothing that is not relevant* for digital libraries. This corresponds to ontological completeness and clarity in (Wand & Weber, 1993), - see below for more details.

How do we know what constitutes the domain of digital libraries? If there existed an established ontology of the digital library domain representing the "real world", one approach would be to compare the constructs of DLRM Vocabulary to those of the domain ontology (real world). Wand and Weber (Wand & Weber, 1993) defined ontological completeness of a grammar[8] as the following:

> *"…, a grammar is ontologically complete if and only if it provides at least one grammatical construct for every ontological construct"*

If not ontologically complete, the grammar is o*ntologically incomplete* and has *construct deficit* (lacks constructs to represent at least one ontological construct), and is thereby not able to express everything in the domain.

Moreover, Wand and Weber defined four variants of ontological completeness and clarity:

1. Total completeness and clarity, when any construct in the language represents one and only one ontological construct. This corresponds to maximal domain appropriateness.

2. Ontological completeness with *construct overload* (unclarity), when some constructs in the language represent more than one ontological construct.

3. Ontological completeness with *construct redundancy* (unclarity), when one ontological construct is represented by *more than one* language construct.

4. Ontological completeness with *construct excess* (unclarity), when the language contains constructs that does not represent any ontological construct. This means that the language allows expressing statements that are not relevant in the domain of discourse, hence the domain appropriateness decreases.

However, since the language we are going to evaluate is derived from DLRM, we can not in this case regard DLRM as our established domain ontology (although being so is clearly one of the ambitions of DLRM). Hence, as I am not aware of any other established ontology for the digital library domain we can not do the analysis solely based on (Wand & Weber, 1993).

While we seem to lack an established formalisation of the digital library domain , the Wand and Weber model is still a suitable framework for discussing gaps between what is possible to express in a modelling language and the domain it is meant to be able to describe. We will therefore refer to it wherever appropriate in the following discussion.

[8] Analogous to ”modelling language” in our case

In addition we follow the suggestion in (J. Krogstie, 2008), using relevant *modelling perspectives* as the starting point, and analyse the degree to which the various perspectives are covered by DLRM Vocabulary.

According to (John Krogstie, 2009) a modelling perspective of a modelling language reflects the core phenomena or concepts the language accommodates, and thereby indicates which types of models it may produce. Conceptual modelling languages of today typically cater for one or more of the following perspectives (examples of core constructs in parentheses) : Structural perspective (entity, object, thing, phenomena), behavioural perspective (transition, state), functional perspective (process, activity, function, etc), Goal and rule perspective (goal, rule, constraint), object perspective (object, class), communication perspective (message, signal), actor and role perspective (actor, agent, role).

In the digital library we judge the structural, functional, goal and rule and actor and role as particularly relevant. In the following, we will discuss how these perspectives are catered for in DLRM Vocabulary.

5.2.1.1 Structural perspective

Here the concern is to represent the static structure of the system being modelled, which is obviously very important for systems like digital libraries. DLRM Vocabulary supports this by offering a number of classes and properties, representing the various resources of the digital library and their properties.

Modelling the structure of a system usually implies some kind of decomposition of the system into smaller parts. The language under scrutiny should be able to support all the basic kinds of decomposition: generalisation, aggregation and membership.

- Generalisation/specialisation: decomposition of a class of objects into increasingly more specific classes. The result of such a decomposition is often referred to as an “is-a hierarchy”. Example: Animal is a generalisation of person; hence person “is-a” animal. In DLRM Vocabulary this is supported by explicitly specified generalisation hierarchies in terms of the Vocabulary Encoding Schemes. By these, generalisation between objects in the described system may be expressed *implicitly*. However, DLRM Vocabulary does not allow for explicitly defining *new* is-a relationships. For example: In the DLRM Function Type Vocabulary Manage Actor is declared as a subclass of Manage Resource. Since the function wcl:RegsiterInWorldCat is modelled as a ManageActor type of function, and wcl:AddTags has a ManageResource functionType the former function is – in one sense – a generalisation of the latter.

- Aggregation: decomposition of an object into its constituent parts. Example: A (physical) book may be seen as an aggregation of cover and sheets. Aggregation of resources is in DLRM Vocabulary supported by the hasPart property which may be specified between any types of resources. While primarily relevant between objects of the same class, it is also possible to model heterogeneous objects, where the object as a whole is of one type, but has parts from another class. Example: A report (Information Object) may have 2 parts (each an Information Object); Executive Summary and Main Part. A Function may have several parts, one being a help file (an Information Object).

- Grouping/membership: Grouping of objects into sets of which the objects are members. Example: a set of books may be included into a certain collection, of which they are all members. A person may be included into a group called "Library staff". In DLRM Vocabulary grouping and membership are supported through the class of ResourceSet, the DLRM Resourse Set Type Vocabulary and the belongTo property of Resources. For example: Registered user NN of Europeana may belong to the Europeana community "Semantic Web".

Granularity and precision

Each class are subdivided further into hierarchies that are represented by the Vocabulary Encoding Schemes, see Appendix 3. Hence, the resources of a digital library may be described/modelled at many levels of granularity. For instance, a function F "Find public documents" may be a pure search function, in which case it can be modelled as a Function with *functionType Search*. On the other hand, F may include both search and browse facilities – or we *may not know* whether F is a search or browse function – in which case it may med described as the more general *Discovery*. Yet another case would be if F in addition to resource discovery also include some extra facilities, e.g. some kind of acquiring mechanism, in which case we would put it down as a Function with *functionType AccessResource*, which is more generic than *Discovery*. Thus DLRM Vocabulary is able to describe structural information about digital libraries at several levels of specificity, and thereby also – to some extent - express vagueness. However, with DLRM Vocabulary we will not be able to distinguish between vagueness and genericity. For example, if a function G in a digital library description/model has functionType Discovery, there is no way of knowing whether this means that G has both search and browse facilities, that the modeller simply didn't know at the time of modelling, or that G has a third kind of facility which is neither search nor browse, but which is judged by the modeller to be a discovery kind of thing. In the latter case, we have a construct deficit (Wand & Weber, 1993).

Scope

The classes denoting resources in DLRM Vocabularies are *Resource* and its subclasses *Information Object*, *Function*, *Policy*, *Quality Parameter* and *Actor*[9].

In this section we focus on *Information Object*, seeing that this is the defining structural element of a digital library. (The other main classes will in due course be discussed under other modelling perspectives)

DLRM Vocabulary provides no real partitioning of the *InformationObject* type. In the original DLRM Information Object is subdivided into subclasses like Metadata, Annotation and View and others, all expressing *roles* that an Information Object may play relative to another resource. These roles are also represented by corresponding properties, e.g. *hasMetadata*, *hasView*. DLRM Vocabulary only includes the properties, not the *InformationObject* subclasses, seeing that those do not constitute persistent types of information objects. An information object may annotate another information object, and thereby be classified as an *Annotation*, but from other viewpoints it is a "regular" information object.

This leaves DLRM Vocabulary with no vocabulary encoding scheme for information object types. However, there exists others, for instance the DCMI Type Vocabulary (Dublin Core Metadata Initiative, 2008b), which may be taken into use for this purpose. The class of *Resource Format* may also be used to express information about the format type of the modelled information objects, e.g. covering media types. Still, it is not clear how one is able to model subtyping of *Information Object* not related to format and/or medium. One much used example of such subtyping is partitioning into genres.

Hence, *Information Object* appears to be an overloaded construct, corresponding to many things that one typically would want to be able to tell apart in a real digital library. To

[9] …and Architectural Component, representing the Architecture domain, which is disregarded in this work.

eliminate the overload one should decide on which subtypes of *Information Object* it is desirable to distinguish between, and establish language constructs to represent these.

5.2.1.2 Functional perspective

In this perspective the focus is on the functional aspects of the digital library system. I DLRM Vocabulary, this is supported by the Function class and its properties, as well as the vocabulary encoding scheme DLRM Function Type Vocabulary. Although providing a relatively rich typology of functions in the form of subclasses of Function, DLRM Vocabulary is weak when it comes to representing *how they work*, that is, operational description in terms of processes or work/activity flows. Only three constructs offer something here:

- *hasPart* may be used to structure functions into subfunctions, tasks into subtasks
- *interactWith* is a property of Function, denoting with which functions a given function interacts. This will result in some kind of flow, but it is not enough to depict processes. Moreover, *interactWith* is intrinsically symmetric, leaving us with no direction for the flow.
- *retrieve*, and the more specific *return* are properties of functions of type *AccessResource* and *Discover*, respectively, denoting the output of the described function. Since these properties can be assigned only to a subset of the functions, there is no general way of expressing output or result of a function in DLRM Vocabulary.

In addition, the property *actOn* denotes the resources on which the described function operates, thus informing us about scope/application area of the function.

A large portion of the function types in DLRM Vocabulary are management functions (*Manage Digital Library*, *Manage Resource*,...) which are natural to represent as processes/workflows. For this we need better support for expressing work processes in general, including

- flow between functions/tasks
- start and stop conditions
- input and output

None of these elements are really possible to express with DLRM Vocabulary. Thus, we conclude that DLRM Vocabulary does have construct deficits concerning the functional perspective.

5.2.1.3 Goal and rule perspective

Modelling languages supporting this perspective focus on modelling goals and rules. In DLRM Vocabulary this is supported by the *Policy* class, its properties and the vocabulary encoding scheme DLRM Policy Type Vocabulary. As for functions, the type hierarchy for policies is fairly comprehensive, providing an expressive “classification scheme” for individual policies, although the policy typology is mainly defined in terms of which types of resources are regulated by the policy.

However, there are no provisions for modelling the policies as such, that is, the *semantics* of the policies. Policies may be more or less specific. Some policies express *rules* influencing the actions of a set of actors. An example of this is the “Unattended children policy” of the Ohio State Library, stating that *“Any child on the premises should be accompanied by an adult who will be responsible for the child’s conduct.”*

Other policies express *goals*, like The Language policy of Europeana, stating that “*As far as possible, the aim is to provide the public with the information they are looking for in their own language”***.**

The general form of rules are

If *condition* then *prescription*

Condition describes the scope of the rule, the conditions in which the rule apply, - whereas *prescription* prescribes what should happen or be done if the rule applies.

As already indicated DLRM Vocabulary does not contain constructs for expressing rules, nor goals. However, it does contain constructs for characterising policies according to their semantics:

- *isEnforced* indicates whether or not the policy is voluntary
- *isExtrinsic* indicates whether or not the policy is imposed from outside or is the result of a social agreement

- *isPrescriptive* indicates whether or not the policy is operationalised

5.2.1.4 Actor and role perspective

This perspective is about actors – human as well as "technical", individual as well as composite – and the roles they play in the described system. In DLRM Vocabulary this is covered by the Actor and Role classes and their properties.

In this area DLRM Vocabulary supports the standard features as they are known from enterprise modelling:

- Composite actors: Actors may be individual actors, groups or communities
- Composite actor decomposition: hasPart makes it possible to represent e.g. organisational hierarchies
- Role of actors: The property *play* connects actors to roles
- Actor profiles: The property *modelledBy* attaches profiles to the actor
- Actor participation: The property *perform* connects actors to functions they perform. However other forms of participation is not possible to express.

5.2.1.5 The quality domain of DLRM

As will be evident from the above, none of the modelling perspectives explicitly mentions *quality* as something to be explicitly represented by separate constructs. On the contrary, if we have a conceptual model of a system, and wish to evaluate the system (i.e. measure its quality or some aspect thereof), we would typically use the conceptual model as input to the evaluation without making the evaluation result a part of the model.

DLRM, and thereby DLRM Vocabulary, has the unique feature that the quality methods and measurements are included into the digital library model. This makes sense, not least because then the quality methods themselves "automatically" will be subject to quality evaluation. However, being able to attach quality parameters to any resource at any level in the digital library (including the quality parameters themselves) does pose some challenges. One challenge is that of aggregation of quality measures; there is no generally applicable method of aggregating quality measures from a lower to a higher level of granularity. For example, how do we "calculate" the usability score of a function solely on the basis of the usability score on each of its parts? Very often it is necessary to evaluate the aggregated function separately. When using DLRM Vocabulary to describe digital libraries, selecting the resources to which quality parameters should be attached must therefore be object of careful consideration.

5.2.1.6 Summing up domain appropriateness

Above we have analysed the domain appropriateness of DLRM Vocabulary, and found several characteristics:

- DLRM Vocabulary supports a comprehensive view of digital libraries: Content, users, functionality, policies and quality.
- Standard hierarchy/decomposition relations are well supported at Resource level: Both generalisation, aggregation and grouping/membership relations may be expressed
- Content (information objects) may be described with less precision than the objects in the other domains. However other typologies exist that may be used for this.
- Policies and functions are describable mainly as “black boxes”. Except for some properties expressing aspects of their inner workings (*actOn*, *interactWith*, *isEnforced*, *isPrescriptive*, a.o.), DLRM Vocabulary does not provide constructs for describing functions as processes, nor policies as goals and rules.
- Actors and Roles are reasonably well supported, though a more generic participation construct than *perform* might be useful.
- The class of *Digital Library* is not defined as a subclass of *Resource*. Hence none of the generic properties defined at *Resource* level may be applied to *Digital Library*. The result is that Digital Library in DLRM Vocabulary represents no more than a virtual aggregation of its actors, users, functions, policies and quality parameters. However, the policies and quality parameters may be used to create a holistic view of the digital library, provided that policies and quality parameters referring to the digital library as a whole are included. This requires that the values of *agreeWith* and *tender* properties include more than the union of policies governing the various types of resources and the union of quality parameters representing some quality aspect of each of the resources.

5.2.2 Participant knowledge appropriateness

Seeing that the criteria *Participant language knowledge appropriateness* and *Knowledge externalisability appropriateness* are closely related, they will be discussed jointly in this section. While the former criterion considers *any* participant's knowledge, the latter is concerned with a specific type of participants - modellers – and their knowledge related to the language. The main difference is that modellers must be able to use their knowledge actively to create new models (i.e. externalise their knowledge by means of the modelling language), whereas other participants may limit themselves to understand models created by others (working knowledge vs. passive knowledge).

The DLRM Vocabulary in its present form only provides a conceptual basis in terms of classes, properties and vocabulary encoding schemes, it has no modelling notation per se. The question here is whether the concepts in DLRM Vocabulary correspond to the way participants in the modelling effort think about digital libraries, and – in case of modellers – the constructs provided are suited to express their knowledge about digital libraries.

The way DLRM Vocabulary decompose the digital library universe into domains have many similarities to enterprise modelling (Wikipedia), so any individual who is familiar with enterprise modelling and the ideas behind, should have few problems with grasping the overall structure of the DLRM Vocabulary. When it comes to detailed modelling within the individual domains, deeper knowledge might be required. However, since the current version of DLRM Vocabulary does not support modelling of processes, rules and goals, no detailed knowledge within these areas is at present necessary for understanding or modelling functions and policies. On the other hand, describing quality parameters does require knowledge about evaluation of systems and services, including evaluation approaches, methods, criteria and measures.

5.2.3 Comprehensibility appropriateness

The comprehensibility criterion is about how easy it is to understand the concepts and notation of the modelling language in question. Since DLRM Vocabulary has no notation, only the understandability of the conceptual basis is relevant. According to (J. Krogstie, 2008) a number of characteristics of the modelling language influence the comprehensibility of the language:

1. *The number of concepts should be limited, and the core concepts should be general rather than specialised:* The number of constructs in the original DLRM is very high with altogether 270 different terms (218 concepts and 52 relations) (DELOS Network of Excellence on Digital libaries, 2007). However, the hierarchical organisation (in terms of generalisation) of the 218 concepts made it possible to reduce the number of core classes in the DLRM Vocabulary drastically, into 12 rather general classes (*Digital Library*, *Information Object*, *Actor*, *Function*, *Policy*, etc, see Appendix 3 for complete overview). The comprehensive hierarchies of terms under some of the main classes are in DLRM Vocabulary realised as type vocabularies. No doubt the size and number of levels in these hierarchies may make them appear complicated. A remedy for this could be to use only the upper part of the hierarchies, thereby exclude the more fine-grained details. This kind of complexity reduction is in fact partly performed in DLRM Vocabulary compared to the original DLRM, see [crossref]. The number of properties is not equally easy to limit, and here the generalisation hierarchy has the opposite effect, since any property defined on a high level is inherited by its descendants. For intstance, 12 different properties are defined at Resource level, all of which are inherited by all subclasses of Resource and added to the properties defined at lower levels. On the other hand, this also implies that a fair amount of properties are general and applicable to all resources.

2. *The concepts of the language should be easily distinguishable from each other:* It is fair to claim that the 12 core constructs (classes) of DLRM Vocabulary are not overlapping, and as such easily distinguishable. In Wand and Weber terms (Wand & Weber, 1993), no construct overload, nor redundancy is found among the classes, at least not at first glance.

   On a more subtle level, however, there may be a redundancy between *Resource Format* and *Information Object*, occurring in the cases when information objects play the role as metadata to other information objects. *Resource Format* is defined to denote the structure of the resource to which it is assigned. It is left to the modeller to define what is meant by structure of the various resource types. However, one example of resource format mentioned is the file format MPEG-21 as resource format for multimedia. Concerning metadata DLRM does not prescribe any specific vocabulary for this. If for instance the vocabulary DCMI Metadata Terms (Dublin Core Metadata Initiative, 2008a) is applied, the property [dcterms:format] would typically be used to denote file format of the described information object. Hence, if not clarified and detailed further, there could be a confusion as to how *Resource Format* should be used compared to specific metadata schemes.

Most of the properties are also distinct, but there are examples of properties that might be easy to confuse. *ExpressedBy* and *describedBy* are both mappings from *Resource* to *Information Object*. Although the distinction is explicitly explained in the DLRM text document, it is easy to imagine that confusion might arise in practical use, and that inconsistent modelling practices might evolve. In Wand and Weber terms, both construct redundancy and construct overload is possible here. The former occurs if, say, *describedBy* is applied both in the (by DLRM) intended meaning and to denote e.g. surrogates of resources[10]. Construct overload occurs for instance if both properties are used to denote descriptions of resources. Another possibly redundant pair of properties are *hasMetadata* (maps information objects to other information objects representing their metadata) and *hasFormat* (maps resources to their resource format), for the reasons explained in the paragraph above.

3. *The language should be flexible in level of detail and level of precision:* Concerning level of detail, very few of the properties in DLRM Vocabulary are obligatory, hence any resource might be described in more or less detail, assigning values to few or many of the properties. Concerning precision level, the flexibility offered is mainly connected to the <resourcetype>Type[11] properties, to which a more or less precise term from the DLRM <resourcetype> Type vocabulary may be assigned. For example: The property *functionType* of a certain function can be given the value *AccessResource*, the more precise *Discovery*, or the even more precise *Browse*, as the case may be. This kind of flexibility propagates further to other properties with other resource types as value range. Example: The property *regulatedBy* of a function (or any other kind of resource) has a policy as value, which in turn may have a more or less precise *policyType* assigned to it. Thus we will claim that DLRM Vocabulary has a fair amount of flexibility both in level of detail and level of precision.

Summing up comprehensibility appropriateness, we conclude that according to the three criteria above, DLRM Vocabulary is reasonably comprehensible. Nevertheless, there are

[10] A typical example would be a picture of a physical object in a museum. Then the picture is a surrogate for the object.

[11] <resourcetype> is to be replaced by ’function’, ’policy’, ’measure’ or ’qualityParameter’

comprehensibility issues in terms of construct redundancy, construct overload and complexity that should be resolved, either through guidelines for use or through clarification and simplification of DLRM Vocabulary as such.

5.2.4 Technical actor interpretation appropriateness (tool appropriateness)

This criterion involves investigating whether the language is appropriately designed for interpretation and automatic handling by computerised tools (‘technical actors’).

As part of this thesis the original DLRM was coded into Protegé and is therefore formally specified in the OWL/RDF formalism. In other words, the semantics of DLRM is defined as far as the semantics of OWL primitives go, including semantics of subclassing. This means that tools supporting OWL is able to perform some OWL-defined reasoning on the DLRM Protegé model. Example: Assume that the function wos:SaveToEndNoteWeb has *functionType Acquire*, whereas the function wos:LookUpIndex has *functionType Browse*. Since both are – directly or indirectly – subclasses of *AccessResource*, an OWL-compliant tool is able to infer that both functions are of type *AccessResource*, and can use this to answer questions like “How many Access Resource functions does Web of Science offer”?

However, as pointed out in previous sections, DLRM does not formally define the semantics of functions and policies, nor completely for quality parameters, although the latter is better prepared for formalisation than are the former two. This implies that DLRM, in its present state of formalisation does not support more advanced automatic reasoning, like deciding whether a set of policies are in conflict. In an evaluation context there are several types of reasoning that would be relevant and very convenient to perform automatically.

Examples:

- Since quality parameters may be attached to resources at any level of detail, it is highly relevant to calculate or infer measurement values of one quality parameter based on the measurement values of other quality parameters. While not able to express such dependencies in the present version of DLRM, being allowed to attach calculation methods to the *affectedBy* mapping between quality parameters would bring us a long way in this direction.

- Some digital libraries are controlled by a multitude of policies, usually in the form of policy documents. Within one digital library it would be highly relevant for instance to identify redundant or conflicting policies. Likewise it is of interest to compare certain policies across digital libraries. For instance: Is the privacy policy of Yahoo! stricter than that of Google? Or wider in scope? Obviously it is no easy matter to automate the reasoning necessary to answer such questions. Rigorously and formally defined semantics of policies are necessary for such tasks.

Based on the above discussion we conclude that the DLRM Vocabulary lend itself relatively easily to tool interpretation and reasoning, at least if we confine ourselves to inference based on the generalisation hierarchy. For more advanced reasoning about complex entities like policy, function and quality parameter however, a more rigorous formalisation of their semantics is necessary.

6 Summary and conclusion

In this thesis I have studied the DELOS Digital Library Reference Model (DLRM) in detail, with the aim to evaluate its suitability as a description profile for search systems, - the ultimate purpose being to apply such descriptions as a basis for comparison and evaluation of search systems.

Two different approaches has been taken to evaluate DLRM. Firstly, a description profile – DLRM Vocabulary – has been derived from the original DLRM (as represented in Protegé) and applied to describe the end-user-accessible functionality in Web of Knowledge, WorldCat Local at State Library of Ohio and Europeana. Secondly, an analytical evaluation of the DLRM Vocabulary as a modeling language has been performed according to an existing quality framework for modeling languages.

The main conclusion is that DLRM Vocabulary does have potential as a description profile for digital libraries. Although our empirical test only comprised the functionality domain, the analytical evaluation indicated that using it will result in a comprehensive view of a digital library, supporting most relevant perspectives, including content, users, functionality, policies and quality. Moreover, the type hierarchy for each resource class makes it possible to distinguish between a number of different types of functions, policies, etc.

No major inconsistencies or conflicts in DLRM Vocabulary are discovered, and the analytical evaluation indicated that it is fairly comprehensible, at least for audience with some knowledge on enterprise modeling. The main problem with using it in its present state is lack of specificity in some areas, causing construct overload or construct deficit.

Below are listed the most important issues identified during evaluation. For more details, consult the appropriate sections in chapters 4 and 5.

- Information Object (representing content) does not have a type hierarchy in DLRM Vocabulary, due to the fact that it is not further subdivided in DLRM (except for subtypes like Query and Collection and role types like Metadata and Annotation). However, there exist other typologies to use for describing information objects.

- DLRM Vocabulary does not allow for process, goal or rule modelling. Therefore functions and policies are hard to formalise to the extent that automatic reasoning is made possible. The property interactWith is meant to represent workflow between functions, but provided that it is symmetric (my interpretation) interactWith does not have any direction..

- Some terms of DLRM Vocabulary are left with unclear or underspecified semantics, which may cause inconsistent modeling. The most prominent example of this is Resource Format, which in the test turned out to be a crucial concept. In addition to being generally unspecified, it is also prone to be confused with Metadata for some resource types.

6.1 Future work

As stated above, my opinion is that DLRM and DLRM Vocabulary has potential for representing digital libraries in a comprehensive and flexible manner. If developed further, to remedy some of the shortcomings pointed out, it should be able to give valuable input to evaluations of digital libraries, especially in cases where a digital library is to be compared to other digital libraries, to some benchmark or standard, or simply to itself at another point in time.

However, to facilitate use of and activity around the reference model, it is absolutely vital that it be published in a suitable machine-readable format, preferably RDF and/or OWL, with a licence that allows experimentation by anyone. I am not aware of the DELOS Association's plans for DLRM, but it is my view that any plans should incorporate formalisation. If so, the Protegé model created during the work with this thesis may be used as a starting point.

It is also important to provide tools and guidelines for managing digital library models created with DLRM Vocabulary, including data registration tools, data analysis tools and possibly data mining tools, - the latter for automatic harvesting of data from the digital libraries themselves. In the test reported here, and for lack of something better, Excel was used for data registration. It is quite clear, however, that this is no viable alternative, even for small projects.

With a suitable infrastructure and tools in place it would be interesting to apply DLRM in further evaluation efforts, possibly in the context of several evaluation paradigms. An alternative close at hand is the Triptych model (

Figure 1) developed within DELOS. By describing the three digital library “cornerstones” system, user and content using the DLRM Vocabulary, the task would be to identify quality parameters to assess the interaction nodes performance, usability and usefulness.

7 References

Ahmed, S. M. Z., McKnight, C., & Oppenheim, C. (2006). A user-centred design and evaluation of IR interfaces. *Journal of Librarianship and Information Science, 38*(3), 157-172.

Bertot, J. C., Snead, J. T., Jaeger, P. T., & McClure, C. R. (2006). Functionality, usability, and accessibility: Iterative user-centered evaluation strategies for digital libraries. *Performance Measurement and Metrics, 7*(1), 17-28.

Blandford, A., Adams, A., Attfield, S., Buchanan, G., Gow, J., Makri, S., et al. (2008). The PRET A Rapporter framework: Evaluating digital libraries from the perspective of information work. *Information Processing & Management, 44*(1), 4-21.

Blandford, A., & Bichanan, G. (2003). Usability of digital libraries. A source of creative tensions with technical developments [Electronic Version]. *TCDL Bulletin*, *1*. Retrieved 2008-11-24, from http://www.ieee-tcdl.org/Bulletin/v1n1/blandford/blandford.html

Blomgren, L., Vallo, H., & Byström, K. (2004). Evaluation of an Information System in an Information Seeking Process. In *Research and Advanced Technology for Digital Libraries* (pp. 57-68).

Borlund, P. (2003). The IIR evaluation model: a framework for evaluation of interactive information retrieval systems. *Information Research - an International Electronic Journal, 8*(3), paper no. 152.

Chowdhury, S., Landoni, M., & Gibb, F. (2006). Usability and impact of digital libraries: a review. *Online Information Review, 30*(6), 656 - 680.

Cleverdon, C., Mills, J., & Keen, E. (1966). *Factors determining the performance of indexingsystems*. Cranfield, UK: Aslib Cranfield Reserach Projecto. Document Number)

DELOS Network of Excellence on Digital libaries. (2007). *The DELOS Digital Library Reference Model. Foundations for Digital Libraries. Version 0.98* o. Document Number)

Dublin Core Metadata Initiative. Retrieved May 25, 2009, from http://dublincore.org/index.shtml

Dublin Core Metadata Initiative. (2007). DCMI Abstract Model. Retrieved May 25, 2009, from http://dublincore.org/documents/2007/06/04/abstract-model/

Dublin Core Metadata Initiative. (2008a). DCMI Metadata Terms. Retrieved May 25, 2009, from http://dublincore.org/documents/dcmi-terms/#H5

Dublin Core Metadata Initiative. (2008b). DCMI Type Vocabulary. Retrieved May 25, 2009, from http://dublincore.org/documents/dcmi-type-vocabulary/

Ellis, D. (1996). The dilemma of measurement in information retrieval research. *Journal of the American Society for Information Science, 47*(1), 23-36.

Europeana Thematic Network. Europeana. Retrieved 24. June, 2009, from http://www.europeana.eu/portal/

Fuhr, N., Hansen, P., Mabe, M., Micsik, A., & Sølvberg, I. (2001). *Digital libraries: A generic classification and evaluation scheme* Paper presented at the European conference on research and advanced technology for digital libraries, 5 (ECDL'01). from http://www.sics.se/~preben/papers/ecdl-2001.pdf

Fuhr, N., Tsakonas, G., Aalberg, T., Agosti, M., Hansen, P., Kapidakis, S., et al. (2007). Evaluation of digital libraries. *International Journal on Digital Libraries, 8*(1), 21-38.

Harman, D. K., & Voorhees, E. M. (2006). TREC: An overview. *Annual Review of Information Science and Technology, 40*, 113-155.

Hider, P. (2006). Search goal revision in models of information retrieval. *Journal of information science, 32*(4), 352-361.

Hornby, A. S. (1994). *Oxford advanced learner's dictionary of current English. 4th. ed.* Oxford, UK: Oxford University Press.

Krogstie, J. (2003). Evaluating UML using a generic quality framework. In *UML and the unified process. Ed. by Liliana Favre* (pp. 1-22): Idea Group Inc (IGI).

Krogstie, J. (2008). Quality of Modelling Languages. In *Model-driven development and evolution of information systems: A quality approach. Draft.* Trondheim: NTNU.

Krogstie, J. (2009). Perspective to modelling. In *Model-driven development and evolution of information systems: A quality approach. Draft.* Trondheim: NTNU.

Krogstie, J., Sindre, G., & Jorgensen, H. (2006). Process models representing knowledge for action: a revised quality framework. *European Journal of Information Systems, 15*(1), 91-102.

Nielsen, J. Ten Usability Heuristics. Retrieved 23. June, 2009, from http://www.useit.com/papers/heuristic/heuristic_list.html

Nielsen, J. (1994). Heuristic evaluation. In J. Nielsen & R. L. Mack (Eds.), *Usability Inspection Methods*. New York, NY: John Wiley & Sons.

OMG. Object Management Group/Business Process Management Initiative. Retrieved 17. June 2009, from http://www.bpmn.org/index.htm

OMG. UML® Resource Page. Retrieved 23. June 2009, from http://www.uml.org/

Pharo, N. (2004). A new model of information behaviour based on the Search Situation Transition schema. *Information Research, 10(1) paper 203*

Pharo, N., & Jarvelin, K. (2004). The SST method: a tool for analysing Web information search processes. *Information Processing & Management, 40*(4), 633-654.

Protegé. (2009). Welcome to Protegé. Retrieved 23. June 2009, from http://protege.stanford.edu

Saracevic, T. (2000). Digital library evaluation: Toward an evolution of concepts. *Library Trends, 49*(2), 350-369.

Saracevic, T. (2004). *Evaluation of digital libraries: An overview.* Paper presented at the DELOS WP7 Workshop on the Evaluation of digital libraries.

Saracevic, T., & Kantor, P. (1997a). Studying the value of library and information services. I. Establishing a theoretical framework. *Journal of the American Society for Information Science, 48*(6), 527-542.

Saracevic, T., & Kantor, P. (1997b). Studying the value of library and information services. II. Methodology and Taxonomy. *Journal of the American Society for Information Science, 48*(6), 543-563.

Sharp, H., Rogers, Y., & Preece, J. (2007). *Interaction design - beyond human-computer interaction. 2nd ed.* Chichester: John Wiley & Sons Ltd.

Speiser, M. (2007). DELOS A Network of Excellence on Digital libaries. Retrieved April 4th, 2009, from http://cordis.europa.eu/ist/digicult/delos.htm

State Library of Ohio. Search State Library of Ohio and beyond [with WorldCat]. Retrieved 24. June, 2009, from http://statelibraryofohio.worldcat.org/

Tague-Sutcliffe, J. M. (1992). The pragmatics of information retrieval experimentation, revisited. *Information Processing & Management, 28*(4), 467-490.

Thomson Reuters. ISI Web of Knowledge. Retrieved 24. June, 2009, from http://www.isiwebofknowledge.com/

Turpin, A., & Scholer, F. (2006). *User performance versus precision measures for simple search tasks.* Paper presented at the SIGIR 06, Seattle, Washington, USA.

Wahl, T., & Sindre, G. (2006). An Analytical Evaluation of BPMN Using a Semiotic Quality Framework. In *Advanced Topics in Database Research. Volume 5. Ed. by Keng Siau* (Vol. 5): Idea Group, Inc.

Wand, Y., & Weber, R. (1993). On the ontological expressiveness of information analysis and design grammar. *Journal of Information Systems, 3*(4), 217-237.

Wikipedia. Enterprise modelling. Retrieved 24. June, 2009, from http://en.wikipedia.org/wiki/Enterprise_modelling

Xie, H. (2006). Evaluation of digital libraries: Criteria and problems from users' perspectives. *Library & Information Science Research, 28*(3), 433-452.

APPENDIX 1 - Encoding of DLRM into Protegé

To facilitate better overview and analysis of DLRM, its salient parts have been encoded into Protegé 4.0, an open source ontology editor based on OWL. Although mainly a straightforward task, the encoding process did pose some challenges, - some due to inconsistencies in the DLRM, some caused by limitations in Protegé/OWL.

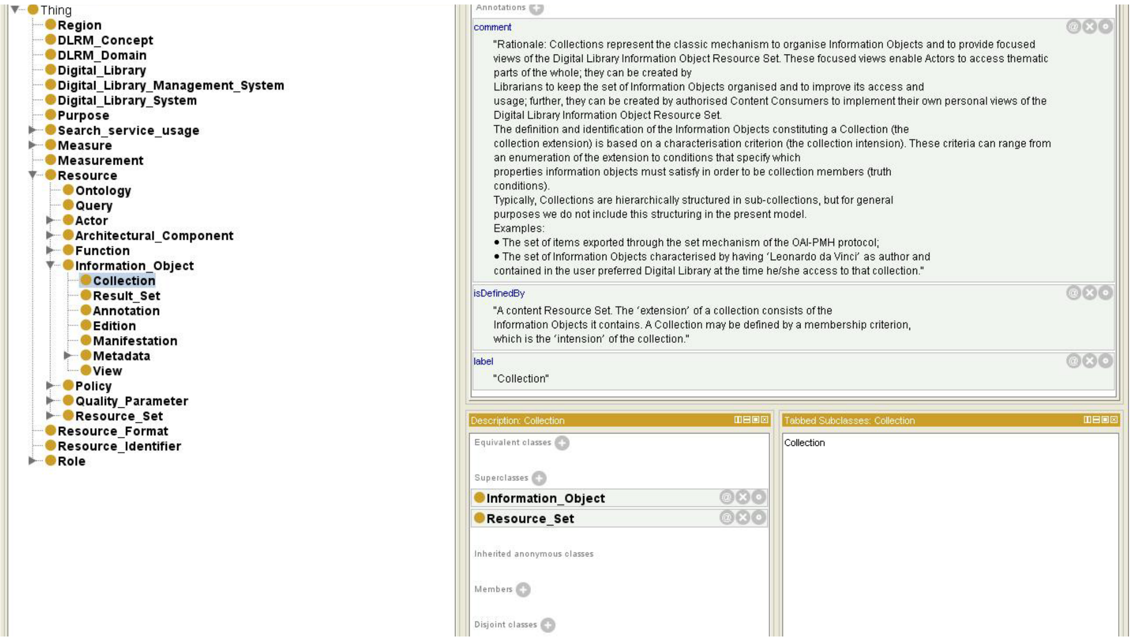
In this chapter the term "DLRM ontology" or only "ontology" refers the DLRM encoded in Protegé. Moreover, "concept" and "relation" refer to entities as defined in the DLRM specification, whereas "class" and "property" refer to their Protegé representations.

Concepts

Entities defined as *concepts* are encoded into Protégé as *classes*, organised hierarchically according to their <isa> relation. Each class is enriched by up to 3 types of annotations:

- label: concept name
- isDefinedBy: the textual definition specified in DLRM
- comment: rationale and examples as specified in DLRM. Additional comments are included as deemed appropriate or necessary.

Figure 18 illustrates how a class is specified. The DLRM concept Collection is represented by the Protegé class Collection, described by three annotations (label, isdefinedBy and Comment) and declared to be a sublass of both Information Object and Resource Set.



comment

"Rationale: Collections represent the classic mechanism to organise Information Objects and to provide focused views of the Digital Library Information Object Resource Set. These focused views enable Actors to access thematic parts of the whole; they can be created by
Librarians to keep the set of Information Objects organised and to improve its access and
usage; further, they can be created by authorised Content Consumers to implement their own personal views of the Digital Library Information Object Resource Set.
The definition and identification of the Information Objects constituting a Collection (the collection extension) is based on a characterisation criterion (the collection intension). These criteria can range from an enumeration of the extension to conditions that specify which
properties information objects must satisfy in order to be collection members (truth conditions).
Typically, Collections are hierarchically structured in sub-collections, but for general purposes we do not include this structuring in the present model.
Examples:
• The set of items exported through the set mechanism of the OAI-PMH protocol;
• The set of Information Objects characterised by having 'Leonardo da Vinci' as author and contained in the user preferred Digital Library at the time he/she access to that collection."

isDefinedBy

"A content Resource Set. The 'extension' of a collection consists of the Information Objects it contains. A Collection may be defined by a membership criterion, which is the 'intension' of the collection."

label

"Collection"

Figure 18 Definition of the Collection class

Classifiers

Classifiers as subdivision criteria

In DLRM some concepts are subdivided into more than one set of subconcepts. That is, several specialisation criteria are used separately, each creating a set of sibling subconcepts spanning the whole (parent) concept. To distinguish between several partitionings of one concept, "special subconcepts" are introduced in DLRM to represent the specialisation criterion used in each case. These "special subconcepts" are referred to as *classifiers* and indicated with brackets in the DLRM document. The best examples of this phenomenon may be found in the Policy Domain, in which the concept of Policy is specialised as follows:

. . C112 Policy
. . . [Policy by characteristic]
. . . . [Policy by context]
. . . . . C113 Extrinsic Policy
. . . . . C114 Intrinsic Policy
. . . . [Policy by expression]
. . . . . C115 Explicit Policy
. . . . . C116 Implicit Policy
…
. . . [Policy by scope]
. . . . C121 System Policy
. . . . C127 Content Policy
. . . . C136 User Policy
. . . . C142 Functionality Policy

As indicated above, each classifier represents one way of subdividing the parent concept (here Policy). For example, [Policy by characteristics] represents a subdivision of the Policy concept based on characteristics of the policy itself, whereas [Policy by scope] represents a subdivision of the Policy concept according to its application domain (i.e. based on the digital library component actually regulated by the policy).

In the DLRM ontology classifiers of this type are represented as annotation to the SubclassOf Axiom. For example, the axiom *System Policy subclassOf Policy* has the annotation *isDefinedBy: "Policy by scope"*

Classifiers as concept domains

In the hierarchical overview of the concepts classifiers are used to structure the hierarchy according to the concept domains of DLRM. Examples are [Content Resource] and [User Resource]. Such classifiers are in effect identical to the consistOf relation between the domains and their concepts, and both the concepts and their domain membership are recorded in the ontology.

Classifiers grouping relations

Classifiers are also used for grouping relations into domains. For example: [Content relations] includes relations connecting objects in the Content domain (information objects). While having no role in an automatic use of the ontology (it is the domain and range specification that govern which kind of objects can be interconnected by the relation) the grouping may constitute valuable information for the human reader. Moreover, since there is no explicit relationship between relations and domains, this information is included as an annotation (comment) of each relation.

Relations

Entities defined as *relations* are encoded into Protégé as *properties*. Domain and Range are specified as classes. Their textual definition is recorded by the isDefinedBy annotation, rationale and examples as a Comment annotation. Additional annotations are included as deemed appropriate or necessary.

In the DLRM document relations are defined in a separate chapter. For each relation its domain, range and rationale is specified. In addition, the concept descriptions lists applicable relations.

Many of the generic relations, that is, those connecting classes which are situated high up in the hierarchy, are listed not only in the description of the domain and range classes, but also in many of their subclasses.

Example:

The relation *govern* connects the class *Policy* (domain) with the class of *Resource* (range), expressing that a policy governs a resource. Hence, the descriptions of the classes of *Policy* and *Resource* both list *govern* as an applicable relation. Moreover, *govern* it is also included in the descriptions of many (but not all) subclasses of *Policy* and *Resource*. This provides valuable information about which types of policies may be governing which types of resources. However, if *govern* was to be encoded merely according to its definition all this information would be lost. To preserve this information in the DLRM ontology, 2 strategies are applied:

1. In cases where the relation in question is listed as applicable between a subclass of the domain and the range (e.g. *Preservation Policy govern Resource*), the domain subclass is explicitly included in the domain specification of the relation. In the case of *govern* this means that both Policy and Preservation Policy is listed as domain, even though Preservation Policy is a specialisation of Policy.

2. In cases where the relation in question is listed as applicable between the domain or one of its subclasses and a subclass of the range (e.g. Digital Rights Management Policy govern Function), a subrelation of the original relation is defined. For our example the relation of *governFunction* connecting *Digital Rights Management Policy* (domain) and *Function* (domain) is defined as a subrelation to *govern*. Once such a new relation is defined it is subject to the same consideration as the original relations, implying that more than one level of subrelations may be defined. Hence, since govern is listed as relating *Change Management Policy* to *Manage Resource* (subclass of *Function*), a new subrelation to *governFunction*, *governManageResource*, is defined for this purpose.[12]

[12] For sake of simplicity, none of these extra relations are included in DLRM Vocabulary

Ternary relations

Most of the relations are binary, connecting objects from the domain to objects from the range. In two cases, however, ternary relations are defined, each specified as a binary relation which itself has a relation to a third class. The two ternary relations are:

- The relation <hasAnnotation> connects a Resource with an Information Object denoting the annotation, which may concern a specific Region (of the resource)

- The relation <associatedWith> may connect two Resources, and the association may have a Purpose attached to it. Note that the purpose is not meant to express the purpose of the related resources, but of the relation between them.

In the following we use <associatedWith> to illustrates the encoding principle used in the DLRM ontology.

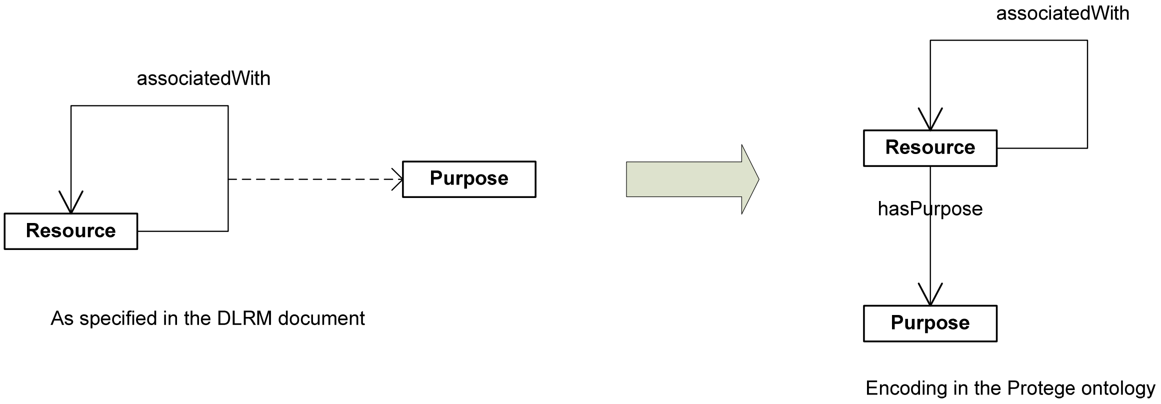


Figure 19 Encoding of the ternary relation <associatedWith>

In UML this may be represented by defining associatedWith as an association class, i.e. a relation that may itself be related to other classes, as depicted in the left side of the figure. However, in Protegé properties (relations) may not be assigned properties. In the Protégé ontology this is solved by

- Defining *associatedWith* as an ordinary binary relation with *Resource* as both domain and range.

- Defining a new relation hasPurpose with *Resource* as domain and *Purpose* as range.

Although this encoding does change the semantics of the associatedWith relation, my opinion is that the effect is minor, which is also illustrated by the following example: An information object (instance of *Information Object*) representing a scientific experiment has an

<associatedWith> relation to an information object representing a scientific paper about the experiment. To express that the purpose of this association is ‘scholarly communication’, a *Purpose* instance to that effect is created and attached to the target of the relation (the information object representing the scientific paper), instead the relation itself.

Domains as concepts vs. organising constructs

In general, the concepts defined in DLRM are entites that may be instantiated into concrete objects in the digital library world, that is, digital libraries, digital library systems, digital library management systems, or component thereof. Examples:

- The concept of Digital Library may be instantiated into a specific digital library, e.g. the ACM Digital Library

- The concept of *Information Object* may be instantiated into the paper by Agichstein etc al., presented at the SIGIR 2006 conference

- The concept of *Metadata* may be instantiated into the metadata record of the above paper in the ACM digital library

- The concept of Search (subconcept of Discover, then Function) may be instantiated into the search function of the ACM Digital Library!

- The concept of Software Component may be instantiated into the program code in the ACM Digital Library system performing the search functions

Domains are another type of entities in DLRM. Although defined and described as “ordinary” concepts in the DLRM, they have a different meaning compared to the other concepts. Domains are constructs defined to impose structure on the reference model itself, meaning they are not to be instantiated into the real world of digital libraries, but to stay – and be used - at the model level.

The DLRM metamodel

To make this distinction easier to see, the metamodel of DLRM is shown in Figure 20. The metamodel of a model M is a model of the constructs used to build M, in other words, the modelling language used to build M. In this case the metamodel is specified as a UML model.

The metamodel shown is inferred from DLRM itself, and modified according to my understanding of how the domains are intended to work, not necessarily how they are actually specified in DLRM. My interpretation is based on two points:

1. The statement made in the DLRM document just in front of the summarized concept hierarchy:

   "This section presents a more formal description of the model in terms of a hierarchy of classes corresponding to the high-level concepts of the current model. This hierarchy does not include the Domain concepts that characterise the Digital Library universe. These are kinds of modules that have been introduced as a way of structuring the model into easily understandable units." (p.69)

2. The UML models screenshots provided as appendices to DLRM. There the domains are modelled as packages, the main organising construct in UML models.

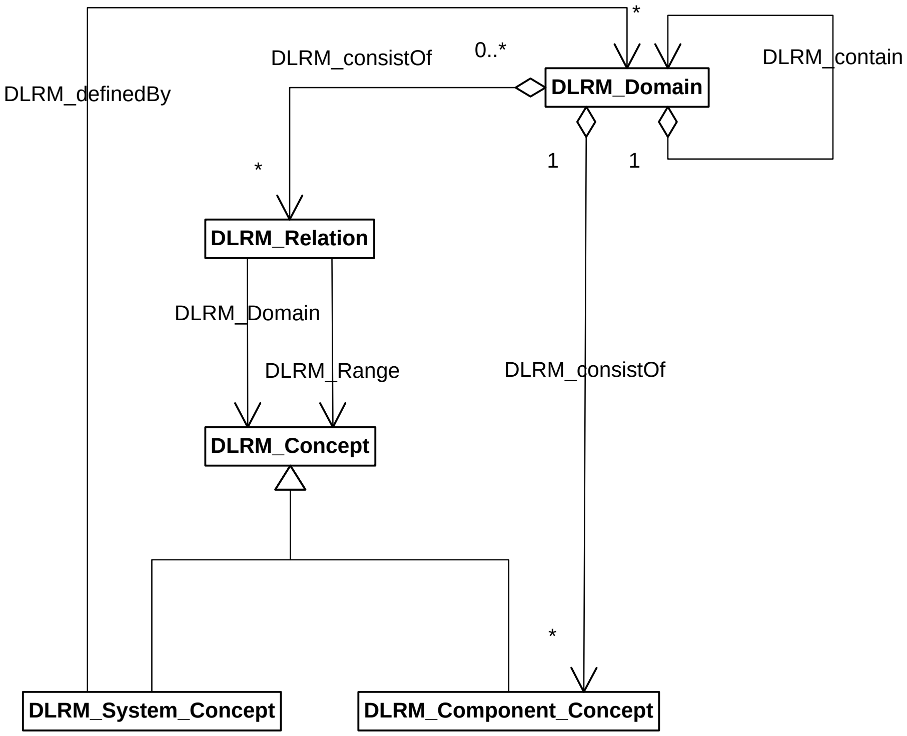


Figure 20 The DLRM metamodel

In order not to confuse the metamodel with DLRM itself, all constructs in the former have the "DLRM_" prefix. As shown in Figure 20, there are 4 main constructs in this language: DLRM_Concept (subdivided into DLRM_System_Concept and DLRM_Component_Concept), DLRM_Relation and DLRM_Domain. Here the container facilities provided by DLRM_Domain is clearly shown by it membership relation to

DLRM_Component_Concept and DLRM_Relation, by its recursive containment relation to itself, and by its ability to define the DLRM:System_Concepts.

Note that representing these relationships at the meta level instead of defining them as instantiations of the DLRM_Relation we have been able to express that each domain (instance of DLRM_Domain) consists of a *certain set of specified concepts*, *not* the real world instantiations of those concepts. Hence, the Content Domain consists of the concepts Information Object and its subconcepts, not specific information objects in the real world. However, this is not how domains are handled in DLRM today, as will be evident from the following..

Below the constructs of the metamodel (Modelling language) is listed together with their corresponding entities in DLRM:

1. DLRM_System_Concept , in DLRM instantiated into digital library and systems concepts (Digital Library, Digital Library System and Digital Library Management System);

2. DLRM_Component_Concept, in DLRM instantiated into other concepts in DLRM (e.g. Resource, Information Object and Function).

3. DLRM_Domain, in DLRM instantiated into one of the 6 domain concepts (User Domain, Content Domain, etc).

4. DLRM_Relation, in DLRM instantiated into relations (actOn, hasMetadata, has Annotation, etc). Also the membership and containment relations from domains to concepts are implicitly specified as ordinary relations, without explicitly indicating that these relations deal with concepts, not with real world objects.

The above shows that, although the metamodel in Figure 20 does seem to capture the intended meaning and usage of domains in the DLRM, DLRM does not in its present state completely comply with this metamodel. The way DLRM handles domains and relations between domains and domains and other concepts is not quite consistent.

For example, consider the following two relation declarations

- Content Domain <consistOf> Information Object (from the description of the concept Content Domain, p. 76)

- Information Object is <regulatedBy> Policy (from the description of the concept Information Object, p. 77)

- Nothing in the text indicated that these statements should be interpreted differently, as they both define a relation type between two concepts. Hence the first one must be taken to mean that an instance of Content Domain may consist of some instance of Information Object, which is clearly not the intended use of domains.

Representing domains in Protegé

To represent domains according to the DLRM metamodel we need one of the following:

- A container (organising) construct in out representation language, and an availability to express relations between packages and other constructs. Protegé/OWL does not provide any of this

- An ability to express relationships between instances and Classes (other than instanceOf). In OWL relationships are assigned between instances (although defined at class level)

As far as I can see there are 2 main paths to follow:

1. Using instances:

    - Defining a class Domain a class Concept and a property consistOf with domain = Domain and range=Concept

    - Generate Domain instances and Concept instances to express the various consistOf relations. The instances of the Concept class are to be interpreted as any concept in the real classes, e.g. Function_Concepts is to be interpreted as Function as any of its subconcepts.

2. Using classes:

- Representing domains as classes, using subdivisioning <isa> as containment relation.

- Defining relationship (properties) DLRM_consistOfConcept between the domain classes and the concept classes. In this case we must impose a special interpretation on the relationship, as one relating the domain not to an instance of the class, but to the class (representing a concept) itself.

The second alternative was chosen for the DLRM ontology, as it seems to imply a "cleaner model" with fewer non-standard interpretations. Note that none of the above will result in a correct model, but will provide valuable information for the human reader.

APPENDIX 2 – DLRM Concepts with properties

The table shows the concepts in DLRM as an intended list, one concept per row. Its properties are shown in yellow cells. The orange cells contain properties of other concepts for which the class on the current row is the range.

Concept				Properties																					
Purpose				is DLRM_consistOfResourceConcept	is hasPurpose																				
Measurement				accordTo	is DLRM_consistOfQualityConcept	is measuredBy																			
Resource				associatedWith	belongTo	describedBy	expressedBy	hasAnnotation	hasFormat	hasMetadata	hasPart	hasPurpose	hasQuality	identifiedBy	regulatedBy	is actOn	is affectedBy	is associatedWith	is create	is DLRM_consistOfResourceConcept	is govern	is hasPart	is implement	is manage	is retrieve
	Architectural_Component			composedBy	conflictWith	hasProfile	implement	use	yield	is composedBy	is conflictWith	is DLRM_consistOfArchitectureConcept	is has	is profile	is use										
		Software_Architecture_Component																							
			Interface																						
	Policy			antonymOf	govern	governFunction	is agreeWith	is antonymOf	is DLRM_consistOfPolicyConcept	is regulatedBy															
		Functionality_Policy																							

			Security_Policy				
			Access_Policy				
				Charging_Policy	governActor		
		System_Policy					
			Connectivity_Policy		governActor		
			Resource_Management_Policy				
			Change_Management_Policy		governManage&ConfigureDLS	governManageDL	governManageResource
			Support_Policy		governActor		
			Risk_Management_Policy				
		Extrinsic_Policy					
		Voluntary_Policy					
		Descriptive_Policy					
		Enforced_Policy					
		Content_Policy					
			Digital_Rights		governInformationObject	is governPolicy	
			Disposal_Policy		governActor		
			Preservation_Policy				
			Collection_Development_Policy		governAccessResource	governActor	governInformationObject
			Collection_Delivery_Policy				
			Digital_Rights_Management_Policy		governManage&ConfigureDLS	governPolicy	
				Licence	grantedTo		
			Submission_and_Resubmission_Policy				
		Prescriptive_Policy					

			Submission_and_Resubmission_Policy												
		Explicit_Policy													
			Submission_and_Resubmission_Policy												
		Implicit_Policy													
		User_Policy			governManageResource										
			Privacy_and_Confidentiality_Policy												
			Acceptable_User_Behaviour_Policy												
			Personalisation_Policy		governActor	governManageActor									
			User_Management_Policy		governActor										
				Registration_Policy	governManageActor										
		Intrinsic_Policy													
			Risk_Management_Policy												
			Submission_and_Resubmission_Policy												
	Actor				belongToGroup	modelledBy	perform	play	is actOnActor	is DLRM_consistOfActorConcept	is expressAssessment	is governActor	is grantedTo	is retrieveActor	is serve
		Group			is belongToGroup										
			Community												
	Function				actOn	influencedBy	interactWith	is DLRM_consistOfFunctionConcept	is governFunction	is interactWith	is offer	is perform	is yield		
		Collaborate			create	retrieve									
			Find_Collaborator		retrieveActor										
			Exchange_Information												
			Author_Collaboratively		createVersio										

									n																					
			Converse																											
		Manage_Resource							create	is governManageResource																				
			Manage_Information_Object						actOnInformationObject																					
				Disseminate																										
					Publish																									
				Author					createInformationObject																					
					Compose																									
				Process																										
					Analyse																									
						Create_Structured_Representation																								
						Qualitative_Analysis			measure																					
							Examine_Preservation_State																							
						Compare																								
						Statistical_Analysis																								
						Linguistic_Analysis																								
						Scientific_Analysis																								
					Transform																									
						Convert_to_a_Different_Format			createView																					
						Extract			createManifestation																					
						Physically_Convert			createManifestation																					
							Translate																							
			Manage_Policy																											
			Annotate						createAnnotation																					
			Validate																											
			Manage_Function																											
			Manage_Quality_Parameter																											
			Manage_Actor						actOnActor																					

				Personalise					is governManage Actor																					
					Apply_Profile																									
				Establish_Actor																										
					Login				is governManage Actor																					
					Register																									
						Sign_Up																								
			Update																											
			Withdraw						actOnResourceSet																					
			Submit						actOnResourceSet																					
			Create																											
		Manage&Configure_DLS																												
			Configure_DLS																											
				Configure_Policy																										
				Configure_Functionality																										
				Configure_Resource_Format																										
				Configure_Content																										
				Configure_User					is governManage&ConfigureDLS																					
				Configure_Quality																										
			Manage_DLS						is governManage&ConfigureDLS																					
				Withdraw_DLS																										
				Manage_Architecture																										
					Monitor_Architectural_Component																									
					Deploy_Architectural_Component																									
					Manage_Architectural_Component																									
					Configure_Architectural_Component																									
				Create_DLS																										
				Update_DLS																										
		Access_Resource							retrieve																					

			Discover						actOnResourceSet	return																				
				Browse					is governAccessResource																					
				Search					issue																					
			Visualise						is governAccessResource																					
			Acquire																											
		Manage_DL							is governManageDL																					
			Manage_User																											
				Manage_Role																										
				Manage_Membership																										
				Manage_Group																										
				Manage_Actor_Profile																										
			Manage_Policy_Domain																											
			Manage_Functionality																											
				Monitor_Usage																										
			Manage_Quality																											
			Manage_Content																											
				Preserve																										
				Manage_Collection																										
					Import_Collection																									
					Export_Collection																									
	Quality_Parameter								affectedBy	evaluatedBy	expressAssessment	measuredBy	is DLRM_consistOfQualityConcept	is hasQuality	is tender															
		Functionality_Quality_Parameter																												
			Robustness																											
			Usability																											
			Availability																											
			Expectations_of_Service																											
			Orthogonality																											
			Impact_of_Service																											
			Fault_Management_Perfor																											

			mance																						
			Awareness_of_Service																						
			User_Satisfaction																						
			Capacity																						
			Dependability																						
		Content_Quality_Parameter		is measure																					
			Perceivability																						
			Fidelity																						
			Trustworthiness																						
			Preservation_Performance																						
			Viability																						
			Authenticity																						
			Size																						
			Scope																						
			Freshness																						
			Metadata_Evaluation																						
			Provenance																						
			Integrity																						
		Generic_Quality_Parameter																							
			Interoperability_Support																						
			Reputation																						
			Documentation_Coverage																						
			Economic_Convenience																						
			Scalability																						
			Sustainability																						
			Security_Enforcement																						
			Performance																						
		User_Quality_Parameter																							
			User_Activeness																						
			User_Behaviour																						
		Policy_Quality_Parameter																							
			Policy_Consistency																						
			Policy_Precision																						
		Architecture_Quality_Parameter																							
			Log_Quality																						
			Redundancy																						
			Ease_of_Installation																						
			Compliance_to_Standards																						
			Load_Balancing_Performance																						
			Ease_of_Administration																						
			Maintenance_Performance																						

	Information_Object			belongToCollection	hasEdition	hasManifestation	hasView	is actOnInformationObject	is createInformationObject	is createManifestation	is createVersion	is createView	is describedBy	is DLRM_consistOfInformationConcept	is expressedBy	is governInformationObject	is hasAnnotation	is hasEdition	is hasManifestation	is hasMetadata	is hasView				
		Annotation		aboutRegion	is createAnnotation	is createAnnotation																			
		Collection		hasExtension	hasIntension	is belongToCollection																			
		Manifestation																							
		View																							
		Query		produce	is hasIntension	is issue																			
		Ontology		is expressionOf																					
		Metadata																							
			Component_Profile	profile	is hasProfile																				
			Actor_Profile	is influencedBy	is modelledBy																				
		Edition																							
		Result_Set		is produce	is return																				
	Resource_Set			is actOnResourceSet	is belongTo	is hasExtension																			
		Collection		hasIntension	is belongToCollection																				
		Group		is belongToGroup																					
			Community																						
		Result_Set		is produ	is return																				

			ce																					
Role			is DLRM_consistOfActorConcept	is play																				
	End-User																							
		Content_Consumer																						
		Librarian																						
		Content_Creator																						
	DL_System_Administrator																							
	DL_Application_Developer																							
	DL_Designer																							
Digital_Library_Management_System			definedBy	deploy	extend	manage																		
Digital_Library_System			definedBy	has	manage	support	is deploy	is extend																
Digital_Library			agreeWith	definedBy	manage	offer	serve	tender	is support															
Search_service_usage																								
	Usage_purpose																							
		Citation																						
		Learning																						
		Entertainment																						
	Information_access																							
		Find_fact																						
		Find_content_type																						
		Find_known_item																						
		Find_about																						
Resource_Format			expressionOf	is DLRM_consistOfResourceConcept	is hasFormat																			
Measure			is accordTo	is DLRM_consistOfQualityConcept	is evaluatedBy																			

	Objective_Measure																													
	Qualitative_Measure																													
	Subjective_Measure																													
	Quantitative_Measure																													
Region									is about Region	is DLRM_consistOfResourceConcept																				
Resource_Identifier									is DLRM_consistOfResourceConcept	is identifiedBy																				

APPENDIX 3 - DLRM Vocabulary specification

Each class is specified with the following attributes:

- URI: The URIs are given as qualified names, that is, [<namespace-prefix>:<term>]. The namespace used for terms originating from concepts and relations in DLRM is assumed to have the prefix “dlrm”. However, at this point no URI for the dlrm namespace is defined. Where deemed appropriate terms from DCMI Metadata terms are used, involving the two namespaces
  - **dcterms**: http://purl.org/dc/terms/
  - **dcmitype**: http://purl.org/dc/dcmitype/
- Label: If the term comes from an existing vocabulary, both the source label and label to be used in this description profile (DLRM Vocabulary) are given
- Definition: Definition of the term. If the term comes from an existing vocabulary, both the source definition and the definition/usage in this descriptin profile (DLRM Vocabulary) are given
- Type of term: Type of term according to the DCMI Abstract model (property, class, vocabulary encoding scheme and syntax encoding scheme)
- Superclass of: Other terms that are subclass of the current
- Subclass of: Other terms that are superclass of the current

Classes

Term name: DigitalLibrary	
URI	[dlrm:DigitalLibrary]
Label	Digital Library
Definition	An organisation, which might be virtual, that comprehensively collects, manages and preserves for the long term rich Information Objects, and offers to its Actors specialised Functions on those Information Objects, of measurable quality, expressed by Quality Parameters, and according to codified Policies.
Type of term	Class
Subclass of	[rdfs:Resource]
Term name: DigitalLibrarySystem	

URI	dlrm:DigitalLibrarySystem
Label	Digital Library System
Definition	A software system based on a given (possibly distributed) Architecture and providing all the Functions required by a particular Digital Library. Actors interact with a Digital Library through the corresponding Digital Library System.
Type of term	Class
Subclass of	[rdfs:Resource]
Term name: Actor	
URI	[dcterm:Agent]
Label (source)	Agent
Label in DLRM Voc.	Actor
Definition (source)	A resource that acts or has the power to act. Comment: Examples of agents include person, organisation and software agent
Definition/usage in DLRM Voc.	A Resource that represents an external entity that interacts with the Digital Library and is identified by a Resource Identifier. Furthermore, it may have at least one Actor Profile and it may belong to at least one Group and be regulated by a set of Policies. An Actor may be characterized by Quality Parameters and may be linked to other Actors.
Type of term	Class
Subclass of	[dlrm:Resource]
Term name: Function	
URI	dlrm:Function
Label	Function
Definition	A particular operation that can be realized on a Resource or Resource Set as the result of the activity of a particular Actor. It is identified by a Resource Identifier. It may be performed by the Actor or it may refer to the respective supporting process of the Digital Library System
Type of term	Class
Subclass of	[dlrm:Resource]
Term name: InformationObject	
URI	[dlrm:InformationObject]
Label	Policy
Definition	The main Resource of the Content Domain. An Information Object is a Resource identified by a Resource Identifier. It must belong to at least one Collection. It may have Metadata, Annotations and multiple Editions, Views, Manifestations. In addition, it may have Quality Parameters and Policies
Usage within DLDP	A condition, rule, term or regulation governing the operation of a Digital Library.
Type of term	Class
Superclass of	[dcterms:BibliographicResource]
Superclass of	[dcterms:PhysicalResource]

Subclass of	[dlrm:Resource]
Term name: Measure	
URI	[dlrm:Measure]
Label	Measure
Definition	A process for computing and assigning a value to a Quality Parameter according to a unit of measurement.
Type of term	Class
Subclass of	[rdfs:Resource]
Comment	The corresponding Measurement is in this test not defined as a class.
Term name: Policy	
URI	[dlrm:Policy]
Label	Policy
Definition	A condition, rule, term or regulation governing the operation of a Digital Library.
Type of term	Class
Superclass of	[dcterms:Policy]
Superclass of	[dcterms:RightsStatement]
Subclass of	[dlrm:Resource]
Subclass of	[rdfs:Resource]
Comment	Deeming from their definitions [dlrm:Policy] is more general than [dcterms:Policy], the latter being defined as “A plan or course of action by an authority, intended to influence and determine decisions, actions and other matters”. [dlrm:Policy] includes both policies defined from outside authorities and those defined inside the digital library.
Term name: QualityParameter	
URI	[dlrm:QualityParameter]
Label	Quality Parameter
Definition	A Resource that indicates, or is linked to, performance or fulfilment of requirements by another Resource. A Quality Parameter is evaluated by (<evaluatedBy>) a Measure, is measured by (<measuredBy>) a Measurement, and expresses the assessment (<expressAssessment>) of an Actor.
Type of term	Class
Subclass of	[dlrm:Resource]
Term name: Resource	
URI	[dlrm:Resource]
Label	Resource
Definition	An identifiable entity in the Digital Library universe
Type of term	Class
Superclass of	Actor [dcterms:Agent]
Superclass of	Function [dlrm:Function]

Superclass of	Information Object [dlrm:InformationObject]
Superclass of	Policy [dlrm:Policy]
Superclass of	Quality Parameter [dlrm:QualityParameter]
Superclass of	Resource Set [dlrm:ResourceSet]
Subclass of	[rdfs:Resource]
Comment	The Resource concept is abstract, in the sense that it cannot be instantiated directly but only through the instantiation of one of its specialisations.
Term name: ResourceFormat	
URI	[dlrm:ResourceFormat]
Label	Resource Format
Definition	A description of the structure of a Resource. May build explicitly on an Ontology or imply an Ontology
Type of term	Class
Superclass of	[dcterms:MediaTypeOrExtent]
Subclass of	[rdfs:Resource]
Comment	Resource Format in DLRM is more general than Media Type Or Extent. However, in the (many) cases where it is appropriate to specify the format of a resource as mediatype, dcterms:MediaTypeOrExtent should be used.
Term name: ResourceIdentifier	
URI	[dlrm:ResourceIdentifier]
Label	Resource Identifier
Definition	A token bound to a resource that distinguishes it from all other Resources within a certain scope, which includes the Digital Library
Type of term	Class
Subclass of	[rdfs:Resource]
Term name: ResourceSet	
URI	[dlrm:ResourceSet]
Label	Resource Set
Definition	A set of Resources, which is in turn a Resource, often defined for some management or application purpose
Type of term	Class
Superclass of	[dcterms:AgentClass]
Superclass of	[dcmitype:Collection]
Subclass of	[rdfs:Resource]
Comment	

Properties

Term name: actOn	
URI	[dlrm:actOn]
Label	Act On
Definition	The relation connecting Functions to Resources on which they operate.
Usage in DLRM Voc.	A resource operated on by a Function
Type of term	Property
Super-property of	
Subpropery of	
Domain within DLRM Voc.	[dlrm:Function]
Range within DLRM Voc.	[dlrm:Resource]
Comment	
Occurrence/cardinality	

Term name: affectedBy	
URI	[dlrm:affectedBy]
Label	Affected By
Definition	The relation connecting Quality Parameters to other Resources that influence their determination.
Usage in DLRM Voc.	A Resource affecting the Quality Parameter
Type of term	Property
Super-property of	
Subpropery of	[dcterms:relation]
Domain within DLRM Voc.	[dlrm:QualityParameter]
Range within DLRM Voc.	[dlrm:Resource]
Comment	
Occurrence/cardinality	0..n

Term name: agreeWith	
URI	[dlrm:agreeWith]
Label	Agree With
Definition	The relation connecting a digital library to the policies with which it agrees
Usage in DLRM Voc.	A Policy with which the digital library agrees
Type of term	Property
Super-property of	
Subpropery of	
Domain within DLRM Voc.	[dlrm:DigitalLibrary]
Range within DLRM Voc.	[dcterms:Policy]
Comment	See also [dlrm:regulatedBy]
Occurrence/cardinality	

Term name: associatedWith	

URI	[dlrm:associatedWith]
Label	Associated With
Definition	The relation connecting a Resource to the Resources that are linked to the former according to a certain Purpose.
Usage in DLRM Voc.	A Resource associated with the Resource for some specific purpose
Type of term	Property
Super-property of	
Subpropery of	[dcterms:relation]
Domain within DLRM Voc.	[dlrm:Resource]
Range within DLRM Voc.	[dlrm:Resource]
Comment	If it were not for the Purpose attached to the association, [dcterms:relation] could serve as the URI for associatedWith
Occurrence/cardinality	
Term name: belongTo	
URI	[dlrm:belongTo]
Label	Belong To
Definition	The relation connecting Resources to the Resource Sets in which they belong. A specialisation of this is the relation connecting Information Objects to the Collections that defines which Collections an Information Object belongs to. Another specialisation of this is the relation connecting an Actor to a Group that defines which user group an actor belongs to.
Usage in DLRM Voc.	A Resource Set in which the Resource belongs
Type of term	Property
Super-property of	
Subpropery of	[dcterms:relation]
Domain within DLRM Voc.	[dlrm:Resource]
Range within DLRM Voc.	[dlrm:ResourceSet]
Comment	
Occurrence/cardinality	0..n
Term name: create	
URI	[dlrm:create]
Label	Create
Definition	The relation connecting the Create Functions to Resources they create. A specialisation of this is the relation connecting the Author Functions to the Information Objects created.
Usage in DLRM Voc.	
Type of term	Property
Super-property of	
Subpropery of	[dcterms:relation]
Domain within DLRM Voc.	[dlrm:Function]
Range within DLRM Voc.	[dlrm:Resource]
Comment	

Occurrence/cardinality	
Term name: describedBy	
URI	[dlrm:describedBy]
Label	Described By
Definition	The relation connecting Resources to Information Objects describing them.
Usage in DLRM Voc.	An Information Object that describes the resource
Type of term	Property
Super-property of	
Subpropery of	[dlrm:hasMetadata]
Domain within DLRM Voc.	[dlrm:Resource]
Range within DLRM Voc.	[dlrm:InformationObject]
Comment	See also [dcterms:description]. The difference between dcterms:description and dlrm:decsribedBy is analogous to the difference between [dcterms:format] and 8dcterms:hasFormat]
Occurrence/cardinality	0..n
Term name: description	
URI	[dcterms:description]
Label	Description
Definition	An account of the resource
Usage in DLRM Voc.	
Type of term	Property
Super-property of	
Subpropery of	
Domain within DLRM Voc.	[dlrm:Resource]
Range within DLRM Voc.	[rdfs:Resource]
Comment	
Occurrence/cardinality	0..n
Term name: evaluatedBy	
URI	[dlrm:evaluatedBy]
Label	Evaluated By
Definition	The relation connecting Quality Parameters to the Measures according to which they are evaluated
Usage in DLRM Voc.	A Measure, i.e. the measuring process by which a Measurement is reached. (The value is stored in measuredBy as a literal)
Type of term	Property
Super-property of	
Subpropery of	
Domain within DLRM Voc.	[dlrm:Quality Parameter]
Range within DLRM Voc.	[dlrm:Measure]
Comment	The number of Measures assigned to a Quality Parameter must for a QP must be greater than or equal to the number of

	Measurements
Occurrence/cardinality	
Term name: expressAssessment	
URI	[dlrm:expressAssessment]
Label	Express Assessment
Definition	The relation connecting Quality Parameters to the Actors who are expressing an assessment of a Resource.
Usage in DLRM Voc.	An Actor whose assessment is represented by the Quality Parameter. May be a human actor or not
Type of term	Property
Super-property of	
Subpropery of	[dcterms:relation]
Domain within DLRM Voc.	[dlrm:Quality Parameter]
Range within DLRM Voc.	[dcTerms:Agent]
Comment	
Occurrence/cardinality	Just one Actor per Quality Parameter (but the actor may of course be a composite one, a group, etc)
Term name: expressedBy	
URI	[dlrm:expressedBy]
Label	Expressed By
Definition	The relation connecting Resources to Information Objects materialising them.
Usage in DLRM Voc.	An Information Object that expresses the resource
Type of term	Property
Super-property of	
Subpropery of	[dcterms:relation]
Domain within DLRM Voc.	[dlrm:Resource]
Range within DLRM Voc.	[dlrm:Information Object]
Comment	Primarily to be used for resources that are abstract in nature, like Policy and Quality Parameter, but can be used for any kind of Resources. Could for instance be used for the relationship between resources and their surrogates.
Occurrence/cardinality	
Term name: functionType	
URI	[dlrm:functionType]
Label	Type Of Function
Definition	The type of function according to the DLRM function typology
Usage in DLRM Voc.	
Type of term	Property
Super-property of	
Subpropery of	
Domain within DLRM Voc.	[dlrm:Function]

Range within DLRM Voc.	[rdfs:Class]
Comment	Use DLRMFunctionType Vocabulary
Occurrence/cardinality	0..n
Term name: govern	
URI	[dlrm:govern]
Label	Govern
Definition	The relation connecting Policies to the Resources they control/govern. It is the inverse relation of <regulatedBy>.
Usage in DLRM Voc.	a Resource governed by the Policy
Type of term	Property
Super-property of	
Subpropery of	[dcterms:relation]
Domain within DLRM Voc.	[dlrm:Policy]
Range within DLRM Voc.	[dlrm:Resource]
Comment	inverse property of regulatedBy
Occurrence/cardinality	0..n
Term name: grantedTo	
URI	[dlrm:grantedTo]
Label	Granted To
Definition	The relation connecting Licences to the Actors to which they are granted
Usage in DLRM Voc.	An Actor being granted the Licence
Type of term	Property
Super-property of	
Subpropery of	[dcterms:relation]
Domain within DLRM Voc.	[dlrm:Policy]
Range within DLRM Voc.	[dcterms:Agent]
Comment	
Occurrence/cardinality	
Term name: hasAnnotation	
URI	[dlrm:hasAnnotation]
Label	Has Annotation
Definition	The relation connecting Resources to Information Objects to add an interpretative value to a certain Region.
Usage in DLRM Voc.	An Information Object annotating the resource
Type of term	Property
Super-property of	
Subpropery of	[dcterms:relation]
Domain within DLRM Voc.	[dlrm:Resource]
Range within DLRM Voc.	[dlrm:InformationObject]
Comment	
Occurrence/cardinality	

Term name: hasFormat	
URI	[dlrm:hasFormat]
Label	Has Format
Definition	The relation connecting a Resource to its Resource Format, which establishes the attributes or properties of the Resource, their types, cardinalities and so on.
Usage in DLRM Voc.	A Resource Format
Type of term	Property
Super-property of	[dcterms:format]
Subpropery of	
Domain within DLRM Voc.	[dlrm:Resource]
Range within DLRM Voc.	[dlrm:ResourceFormat]
Comment	Note: [dcterms:hasFormat] is not the same as [dlrm:hasFormat]. The former points to a resource which which is substantially the same as the original, but in another format. The latter points to the format itself, as [dcterms:format] does.
Occurrence/cardinality	
Term name: hasMetadata	
URI	[dlrm:hasMetadata]
Label	Has Metadata
Definition	The relation connecting Resources to Information Objects for management purposes.
Usage in DLRM Voc.	
Type of term	Property
Super-property of	[dlrm:describedBy]
Subpropery of	[dcterms:relation]
Domain within DLRM Voc.	
Range within DLRM Voc.	
Comment	
Occurrence/cardinality	
Term name: hasPart	
URI	[dcterms:hasPart]
Label	Has Part
Definition	A related resource that is included either physically or logically i the described resource
Usage in DLRM Voc.	The relation connecting Resources to their constituent Resources.
Type of term	Property
Super-property of	
Subpropery of	[dcterms:relation]
Domain within DLRM Voc.	
Range within DLRM Voc.	

Comment	
Occurrence/cardinality	
Term name: hasPurpose	
URI	[dlrm:hasPurpose]
Label	Has Purpose
Definition	The motivation characterising the <associatedWith> relationship.
Usage in DLRM Voc.	The relation connecting a Purpose to a Resource that is associatedWith another Resource. The Purpose is related to the association.
Type of term	Property
Super-property of	
Subpropery of	
Domain within DLRM Voc.	[dlrm:Resource]
Range within DLRM Voc.	[dlrm:Purpose]
Comment	
Occurrence/cardinality	
Term name: hasQuality	
URI	[dlrm:hasQuality]
Label	Has Quality
Definition	The relation connecting a Resource to its Quality Parameters.
Usage in DLRM Voc.	
Type of term	Property
Super-property of	
Subpropery of	[dcterms:relation]
Domain within DLRM Voc.	[dlrm:Resource]
Range within DLRM Voc.	[dlrm:QualityParameter]
Comment	
Occurrence/cardinality	0..n
Term name: identifiedBy	
URI	[dlrm:identifiedBy]
Label	Identified By
Definition	The relation connecting a Resource to its Resource Identifier
Usage in DLRM Voc.	
Type of term	Property
Super-property of	
Subpropery of	
Domain within DLRM Voc.	[dlrm:Resource]
Range within DLRM Voc.	[dlrm:ResourceIdentifier]
Comment	See also [dcterms:identifier]. Although similar in meaning and purpose, [dcterms:identifier] has rdfs:Literal as Range, whereas identifiedBy points to a non-literal. Should be integrated in the

	future.
Occurrence/cardinality	
Term name: influencedBy	
URI	[dlrm:influencedBy]
Label	Influenced By
Definition	The relation connecting Functions to Actor Profiles that expresses the fact that Functions are influenced by specific user characteristics
Usage in DLRM Voc.	
Type of term	Property
Super-property of	
Subpropery of	[dcterms:relation]
Domain within DLRM Voc.	[dlrm:Function]
Range within DLRM Voc.	[dlrm:InformationObject] (ActorProfiles)
Comment	
Occurrence/cardinality	
Term name: interactWith	
URI	[dlrm:interactWith]
Label	Interact With
Definition	The relation connecting Functions to Functions that expresses the interaction between them.
Usage in DLRM Voc.	
Type of term	Property
Super-property of	
Subpropery of	[dcterms:relation]
Domain within DLRM Voc.	[dlrm:Function]
Range within DLRM Voc.	[dlrm:Function]
Comment	
Occurrence/cardinality	0..n
Term name: isEnforced	
URI	[dlrm:isEnforced]
Label	Is Enforced
Definition	Indicates that the policy is deployed and strictly applied within the DL. The opposite means that the policy is not deployed or applying it is up to the Actor's own choice.
Usage in DLRM Voc.	
Type of term	Property
Super-property of	
Subpropery of	
Domain within DLRM Voc.	[dlrm:Policy]
Range within DLRM Voc.	[rdfs:#Literal]
Comment	True or False

Occurrence/cardinality	0..1
Term name: isExplicit	
URI	[dlrm:isExplicit]
Label	Is Explicit
Definition	Indicates that the policy has been stated and approved, as opposed to being inherent in the DL, but not documented
Usage in DLRM Voc.	
Type of term	Property
Super-property of	
Subpropery of	
Domain within DLRM Voc.	[dlrm:Policy]
Range within DLRM Voc.	[rdfs:#Literal]
Comment	True or False
Occurrence/cardinality	0..1
Term name: isExtrinsic	
URI	[dlrm:isExtrinsic]
Label	Is Extrinsic
Definition	Indicates that the policy is imposed by a body outside the Digital Library, as opposed to being defined within the DL
Usage in DLRM Voc.	
Type of term	Property
Super-property of	
Subpropery of	
Domain within DLRM Voc.	[dlrm:Policy]
Range within DLRM Voc.	[rdfs:#Literal]
Comment	True or False
Occurrence/cardinality	0..1
Term name: isPrescriptive	
URI	[dlrm:isPrescriptive]
Label	Is Prescriptive
Definition	Indicates that the policy directly constains or manages interactions with the DL, as opposed to providing a mer explanation (description) of the policy
Usage in DLRM Voc.	
Type of term	Property
Super-property of	
Subpropery of	
Domain within DLRM Voc.	[dlrm:Policy]
Range within DLRM Voc.	[rdfs:#Literal]
Comment	True or False
Occurrence/cardinality	0..1

Term name: issue	
URI	[dlrm:issue]
Label	Issue
Definition	The relation connecting Search Functions to the Queries they use to retrieve results.
Usage in DLRM Voc.	A Query issued by some search function
Type of term	Property
Super-property of	
Subpropery of	[dcterms:relation]
Domain within DLRM Voc.	[dlrm:Function] (Search)
Range within DLRM Voc.	[dlrm:InformationObject] (Query)
Comment	
Occurrence/cardinality	0..n

Term name: manage	
URI	[dlrm:manage]
Label	Manage
Definition	The relation connecting a digital library to the resources it manages
Usage in DLRM Voc.	A resource (of any kind)managed by the digital library
Type of term	Property
Super-property of	
Subpropery of	
Domain within DLRM Voc.	dlrm:DigitalLibrary
Range within DLRM Voc.	[dlrm:Resource]
Comment	
Occurrence/cardinality	0..n

Term name: measuredBy	
URI	[dlrm:measuredBy]
Label	Measured By
Definition	The relation connecting Quality Parameters to the Measurements that assign them a value.
Usage in DLRM Voc.	A literal value giving the actual value of the Quality Parameter according to some measuring process (a Measure)
Type of term	Property
Super-property of	
Subpropery of	
Domain within DLRM Voc.	[dlrm:QualityParameter]
Range within DLRM Voc.	[dlrm:Measurement] (in DLRM Vocabulary: a Literal value)
Comment	Literal value for simplicity in this test
Occurrence/cardinality	0..1

Term name: measureType	
URI	[dlrm:measureType]
Label	Type of Measure

Definition	The type of quality measure according to the DLRM Measure typology
Usage in DLRM Voc.	
Type of term	Property
Super-property of	
Subpropery of	
Domain within DLRM Voc.	[dlrm:Measure]
Range within DLRM Voc.	[rdfs:Class]
Comment	Use DLRMMeasureTypeVocabulary
Occurrence/cardinality	0..n
Term name: offer	
URI	[dlrm:offer]
Label	Offer
Definition	The relation connecting a digital library to the Functions it offers to its users
Usage in DLRM Voc.	A Function offered by the digital library
Type of term	Property
Super-property of	
Subpropery of	
Domain within DLRM Voc.	[dlrm:DigitalLibrary]
Range within DLRM Voc.	[dlrm:Function]
Comment	
Occurrence/cardinality	
Term name: policyType	
URI	[dlrm:policyType]
Label	Type Of Policy
Definition	The type of policy according to the DLRM policy typology
Usage in DLRM Voc.	
Type of term	Property
Super-property of	
Subpropery of	
Domain within DLRM Voc.	[dlrm:POlicy]
Range within DLRM Voc.	[rdfs:Class]
Comment	Use DLRMPolicyType Vocabulary
Occurrence/cardinality	0..n
Term name: produce	
URI	[dlrm:produce]
Label	produce
Definition	The relation connecting Queries to Result Sets they characterise
Usage in DLRM Voc.	The result set produced by a Query
Type of term	Property
Super-property of	

Subpropery of	[dcterm:relation]
Domain within DLRM Voc.	[dlrm:InformationObject] (Query)
Range within DLRM Voc.	[dlrm:ResourceSet] (ResultSet)
Comment	
Occurrence/cardinality	
Term name: qualityParameterType	
URI	[dlrm:qualityParameterType]
Label	Type Of Quality Parameter
Definition	The type of quality parameter according to the DLRM quality parameter typology
Usage in DLRM Voc.	
Type of term	Property
Super-property of	
Subpropery of	
Domain within DLRM Voc.	[dlrm:QualityParameter]
Range within DLRM Voc.	[rdfs:Class]
Comment	Use DLRMQualityParameterType Vocabulary
Occurrence/cardinality	0..n
Term name: regulatedBy	
URI	[dlrm:regulatedBy]
Label	Regulated By
Definition	The relation connecting Resources to the Policies regulating them.
Usage in DLRM Voc.	A Policy governing the Resource
Type of term	Property
Super-property of	[dcterms:rights]
Subpropery of	[dcterms:relation]
Domain within DLRM Voc.	[dlrm:Resource]
Range within DLRM Voc.	[dcterms:Policy]
Comment	
Occurrence/cardinality	0..n
Term name: retrieve	
URI	[dlrm:retrieve]
Label	Retrieve
Definition	The relation connecting Access Resource Functions to Resources they find. A specialisation of this relation connects Find Collaborator Functions to Actors they find. Another specialisation is the relation <return> which connects the Function Discover and Result Set.
Usage in DLRM Voc.	A resource retrieved by an access function
Type of term	Property
Super-property of	

Subpropery of	[dcterms:relation]
Domain within DLRM Voc.	[dlrm:Function] (AccessResource)
Range within DLRM Voc.	[dlrm:Resource]
Comment	
Occurrence/cardinality	0..n
Term name: return	
URI	[dlrm:return]
Label	Return
Definition	The relation connecting Discover Functions to Result Sets they find. It is a specialisation of the <retrieve> relation connecting Access Resource Functions to Resources.
Usage in DLRM Voc.	
Type of term	Property
Super-property of	
Subpropery of	[dlrm:retrieve]
Domain within DLRM Voc.	[dlrm:Function] (Discover)
Range within DLRM Voc.	[dlrm:ResourceSet] (ResultSet)
Comment	
Occurrence/cardinality	0..1
Term name: serve	
URI	[dlrm:serve]
Label	Serve
Definition	The relation connecting a digital library to the Actors it serves
Usage in DLRM Voc.	An Actor served by the digital library
Type of term	Property
Super-property of	
Subpropery of	
Domain within DLRM Voc.	[dlrm:DigitalLibrary]
Range within DLRM Voc.	[dcterm:Agent]
Comment	
Occurrence/cardinality	0..n
Term name: isSupportedBy	
URI	[dlrm:support]
Label	[dlrm:isSupportedBy]"
Definition	Support/Is Supported By
Usage in DLRM Voc.	The relation connecting Application Frameworks to the Running Components that support the operation (from DLRM document). Here: The relation connecting the Digital Library to the siftware system that supports it and yield its functions
Type of term	The software system
Super-property of	Property
Subpropery of	

Domain within DLRM Voc.	
Range within DLRM Voc.	[dlrm:DigitalLibrary]
Comment	[dlrm:DigitalLibrarySystem]
Occurrence/cardinality	0..1
Term name: tender	
URI	[dlrm:tender]
Label	Tender
Definition	The relation connecting a digital library to the Quality Parameters it tenders
Usage in DLRM Voc.	A Quality Parameter tendered by the digital library
Type of term	Property
Super-property of	
Subpropery of	
Domain within DLRM Voc.	[dlrm:DigitalLibrary]
Range within DLRM Voc.	[dlrm:QualityParameter]
Comment	In DLRM, only resources can be linked to the Quality Parameter by the hasQuality relation. Hence, Digital Library, not being a dlrm:Resource, can not be assigned Quality Parameter through the hasQuality property. Thus, we interpret the tender property as a reference to the collection of the Quality Parameters linked to any of the digital library' resources.
Occurrence/cardinality	0..n
Term name: title	
URI	[dcterms:title]
Label	Title
Definition	A name given to the resource
Usage in DLRM Voc.	
Type of term	Property
Super-property of	
Subpropery of	
Domain within DLRM Voc.	[dlrm:Resource]
Range within DLRM Voc.	[rdfs:Resource]
Comment	
Occurrence/cardinality	0..n
Term name: type	
URI	[dcterms:type]
Label	Type
Definition	The nature or genre of the resource
Usage in DLRM Voc.	
Type of term	Property

Super-property of	
Subpropery of	
Domain within DLRM Voc.	[dlrm:Resource]
Range within DLRM Voc.	[rdfs:Class]
Comment	Use DCMIType Vocabulary with DLRMType additions
Occurrence/cardinality	

Vocabulary encoding schemes

Term name: DLRMFunctionType	
URI	[dlrm:DLRMFunctionType]
Label	DLRM Function Type Vocabulary
Definition	A set of classes specified by the DLRM Function Type Vocabulary, used to categorize resources that are functions
Type of term	[dcam:VocabularyEncodingScheme]
Namespace prefix for vocabulary terms	dlrmftype
Term name: DLRMMeasureType	
URI	[dlrm:MeasureType]
Label	DLRM Measure Type Vocabulary
Definition	A set of classes specified by the DLRM Measure Type Vocabulary, used to categorize resources that are Measure
Type of term	[dcam:VocabularyEncodingScheme]
Namespace prefix for vocabulary terms	dlrmmtype
Term name: DLRMPolicyType	
URI	[dlrm:DLRMPolicyType]
Label	DLRM Policy Type Vocabulary
Definition	A set of classes specified by the DLRM Policy Type Vocabulary, used to categorize resources that are policies
Type of term	[dcam:VocabularyEncodingScheme]
Namespace prefix for vocabulary terms	dlrmptype
Term name: DLRMQualityParameterType	
URI	[dlrm:DLRMQualityParameterType]
Label	DLRM Quality Parameter Type Vocabulary
Definition	A set of classes specified by the DLRM Quality Parameter Type Vocabulary, used to categorize resources that are quality parameters
Type of term	[dcam:VocabularyEncodingScheme]
Namespace prefix for vocabulary terms	dlrmqptype

Term name: DLRMType	
URI	[dlrm:DLRMType]
Label	DLRM Type Vocabulary
Definition	A set of classes specified by the DLRM Type Vocabulary, used as an addition to DCMI Type Vocabulary, to categorize the nature or genre of the resource
Type of term	[dcam:VocabularyEncodingScheme]
Namespace prefix for vocabulary terms	dlrmtype

DLRM Function Type Vocabulary

The DLRM Function Type Vocabulary contains terms denoting types of functions typically offered by a digital library. The terms correspond to specialisations of the Function concept in DLRM. With two exception only the direct descendants of Function are included. However, for AccessResource the whole hierarchy is included, this being particularly relevant in the context of this work. For ManageResource the subclasses corresponding to the various types of resources are included.

Term	URI	Label	Definition	Type of term	Subclass of	Member Of
AccessResource	[dlrmftype: AccessResource]	Acess Resource	A function which provide Actors with mechanisms for discovering and accessing Resources	Class	[dlrm:Function]	DLRM Function Type Vocabulary
Acquire	[dlrmftype: Acquire]	Acquire	An Access Resource function supporting an Actor in retaining Resources in existence past the lifetime of the Actor's interaction with the system.	Class	[dlrmftype: AccessResource]	DLRM Function Type Vocabulary
Browse	[dlrmftype: Browse]	Browse	An Access Resource function that lists Resources in a Resource Set ordered or organised according to a given characteristic or scheme.	Class	[dlrmftype: Discover]	DLRM Function Type Vocabulary
Collaborate	[dlrmftype: Collaborate]	Collaborate	A function that supports Actors in sharing information, working and	Class	[dlrm:Function]	DLRM Function Type Vocabulary

Term	URI	Label	Definition	Type of term	Subclass of	Member Of
			communicating effectively and efficiently with peers.			
Discover	[dlrmftype: Discover]	Discover	A function which finds a Resource (which may be an individual one or a Resource Set) compliant with the specification of the Actor request, as expressed by a Query or by browsing.	Class	[dlrmftype: AccessResource]	DLRM Function Type Vocabulary
Manage&ConfigureDLS	[dlrmftype: Manage&ConfigureDLS]	Manage and Configure DLS	A function that supports the management and configuration of the DLS that implements the DL.	Class	[dlrm:Function]	DLRM Function Type Vocabulary
ManageActor	[dlrmftype: ManageActor]	Manage Actor	A Manage Resource function supporting the administration of the set of Actors that access the digital library.	Class	[dlrmftype: ManageResource]	DLRM Function Type Vocabulary

Term	URI	Label	Definition	Type of term	Subclass of	Member Of
ManageDL	[dlrmftype: ManageDL]	Manage DL	A function managing the Content, Actors or other Resources of the DL in order in order to achieve the desired Quality Parameters in agreement with the established Policies.	Class	[dlrm:Function]	DLRM Function Type Vocabulary
ManageFunction	[dlrmftype: ManageFunction]	Manage Function	A Manage Resource supporting the administration of the features of the Functions provided by the DL.	Class	[dlrmftype: ManageResource]	DLRM Function Type Vocabulary
Manage Information Object	[dlrmftype: ManageInformation Object]	Manage Information Object	The class of Functions that support the production, withdrawal, update, publishing and processing of Information Objects.	Class	[dlrmftype: ManageResource]	DLRM Function Type Vocabulary
ManagePolicy	[dlrmftype: ManagePolicy]	Manage Policy	A Manage Resource supporting the administration of the set of Policies governing the DL and its Resources.	Class	[dlrmftype: ManageResource]	DLRM Function Type Vocabulary
ManageQuality Parameter	[dlrmftype: ManageQuality Parameter]	Manage Quality Parameter	A Manage Resource supporting the administration of the individual Quality	Class	[dlrmftype: ManageResource]	DLRM Function Type Vocabulary

Term	URI	Label	Definition	Type of term	Subclass of	Member Of
			Parameters, which refer to all aspects of the DL.			
ManageResource	[dlrmftype: ManageResource]	Manage Resource	A function which supports the production, withdrawal or update of Resources	Class	[dlrm:Function]	DLRM Function Type Vocabulary
Search	[dlrmftype:Search]	Search	An Access Resource function that allows an Actor to discover the Resources matching a Query, which are returned as a Result Set. Search must be triggered by a Query.	Class	[dlrmftype: Discover]	DLRM Function Type Vocabulary
Visualise	[dlrmftype:Visualise]	Visualise	An Access Resource function enabling an Actor to view a Resource graphically, such as an Information Object or an Actor Profile.	Class	[dlrmftype: AccessResource]	DLRM Function Type Vocabulary

DLRM Measure Type Vocabulary

The DLRM Function Type Vocabulary contains terms denoting types of measuring methods/procedures applied to assign values/measurements to quality parameters. The terms correspond to specialisations of the Measure concept in DLRM.

Term	URI	Label	Definition	Type of term	Subclass of	Member Of
ObjectiveMeasure	[dlrmmtype: ObjectiveMeasure]	Objective Measure	A Measure obtained via a well defined process that does not depend on individual perception	Class	[dlrm:Measure]	DLRM Measure Type Vocabulary
QualitativeMeasure	[dlrmmtype: QualitativeMeasure]	Qualitative Measure	A Measure based on unit of measurement which is not expressed via numerical values	Class	[dlrm:Measure]	DLRM Measure Type Vocabulary
QuantitativeMeasure	[dlrmmtype: QuantitativeMeasure]	Quantitative Measure	A Measure based on unit of measurement which is expressed via numerical values	Class	[dlrm:Measure]	DLRM Measure Type Vocabulary
SubjectiveMeasure	[dlrmmtype: SubjectiveMeasure]	Subjective Measure	A Measure based on, or influenced by, personal feelings, tastes or opinions	Class	[dlrm:Measure]	DLRM Measure Type Vocabulary

DLRM Policy Type Vocabulary

The DLRM Policy Type Vocabulary contains terms denoting types of policies typically maintained or adhered to by a digital library. The terms correspond to specialisations of the Policy concept in DLRM. Only a subset of the direct descendants of Policy are included in the vocabulary. The rest of the Policy subclasses may be grouped into pairs of antonyms (e.g. Explicit Policy vs Implicit Policy), which are more conveniently included in DLRM Vocabulary as Boolean properties of Policy resources, see 0 above.

Term	URI	Label	Definition	Type of term	Subclass of	Member Of
FunctionalityPolicy	[dlrmptype: FunctionalityPolicy]	Functionality Policy	A Policy regulating the Functionality domain	Class	[dlrm:Policy]	DLRM Policy Type Vocabulary
SystemPolicy	[dlrmptype: SystemPolicy]	System Policy	A Policy that concerns an aspect of a system as a whole, be it a Digital Library, a Digital Library System or a Digital Library Management System	Class	[dlrm: Policy]	DLRM Policy Type Vocabulary
ContentPolicy	[dlrmptype: ContentPolicy]	Content Policy	Policy regulating the Content domain	Class	[dlrm: Policy]	DLRM Policy Type Vocabulary
UserPolicy	[dlrmptype:UserPolicy]	User Policy	Policy regulating the User domain	Class	[dlrm: Policy]	DLRM Policy Type Vocabulary

DLRM Quality Parameter Type Vocabulary

The DLRM Quality Parameter Type Vocabulary contains terms denoting types of quality parameters typically relevant for digital libraries and their resources. The terms correspond to specialisations of the Quality Parameter concept in DLRM. With one exception, only the direct descendants of Quality Parameter are included. Generic Quality Parameters are however subdivided further, as this category seems to contain the most relevant parameters for digital libraries as a whole.

Term	URI	Label	Definition	Type of term	Subclass of	Member Of
ArchitectureQuality Parameter	[dlrmqptype:ArchitectureQualityParameter]	Architecture Quality Parameter	A Quality Parameter that concerns an aspect of the Architecture Domain	Class	[dlrm:QualityParameter]	DLRM Quality Parameter Type Vocabulary
Availability	[dlrmqptype:Availability]	Availability	The Functionality Quality Parameter which indicates the ratio of the time a Functions is ready for use to the total lifetime of the system.	Class	[dlrmqptype:FunctionalityQualityParameter]	DLRM Quality Parameter Type Vocabulary
AwarenessOfService	[dlrmqptype:AwarenessOfService]	Awareness Of Service	The Functionality Quality Parameter which measures how well the perspectives Actor of a Digital Library are aware of its existence and unctions.	Class	[dlrmqptype:FunctionalityQualityParameter]	DLRM Quality Parameter Type Vocabulary
Capacity	[dlrmqptype:Capacity]	Capacity	The Functionality Quality Parameter measuring the limit on the number of requests that a Function can serve in a given interval of	Class	[dlrmqptype:FunctionalityQualityParameter]	DLRM Quality Parameter Type Vocabulary

Term	URI	Label	Definition	Type of term	Subclass of	Member Of
			time.			
ContentQuality Parameter	[dlrmqptype:ContentQualityParameter]	Acess Resource	A Quality Parameter that concerns an aspect of the Content (the information Objects in the DL)	Class	[dlrm:QualityParameter]	DLRM Quality Parameter Type Vocabulary
Dependability	[dlrmqptype:Dependability]	Dependability	The Functionality Quality Parameter measuring the ability of a DL to perform a Function under stated conditions for a specified period of time.	Class	[dlrmqptype:FunctionalityQualityParameter]	DLRM Quality Parameter Type Vocabulary
Documentation Coverage	[dlrmqptype:DocumentationCoverage]	Documentation Coverage	The Generic Quality Parameter measuring the accuracy and clarity of the documentation describing a given Resource	Class	[dlrmqptype:GenericQualityParameter]	DLRM Quality Parameter Type Vocabulary
EconomicConvenience	[dlrmqptype:EconomicConvenience]	Economic Convenience	The General Quality Parameter which reflects how much favourable is the economic efficiency when using of a Digital Library	Class	[dlrmqptype:GenericQualityParameter]	DLRM Quality Parameter Type Vocabulary
ExpectationOfService	[dlrmqptype:ExpectationOfService]	Expectation Of Service	The Functionality Quality Parameter measuring what Actors	Class	[dlrmqptype:FunctionalityQualityParameter]	DLRM Quality Parameter Type Vocabulary

Term	URI	Label	Definition	Type of term	Subclass of	Member Of
			believe a Function should offer.			
FaultManagement Performance	[dlrmqptype:FaultManagementPerformance]	Fault Management Performance	The Functionality Quality Parameter measuring the ability of a Function to re-act to and recover from failures in a transparent way.	Class	[dlrmqptype:FunctionalityQualityParameter]	DLRM Quality Parameter Type Vocabulary
Functionality QualityParameter	[dlrmqptype:FunctionalityQualityParameter]	Collaborate	A Quality Parameter that concerns an aspect of the Functionality of the DL	Class	[dlrm:QualityParameter]	DLRM Quality Parameter Type Vocabulary
GenericQuality Parameter	[dlrmqptype:GenericQualityParameter]	Generic Quality Parameter	A Quality Parameter that concerns an aspect of a "system" as a whole, being it a Digital Library, a Digital Library System, or a Digital Library Management System	Class	[dlrm:QualityParameter]	DLRM Quality Parameter Type Vocabulary
ImpactOfService	[dlrmqptype:ImpactOfService]	Impact Of Service	The Functionality Quality Parameter measures the influence which the service offered by a Function have on the Actor knowledge and behaviour.	Class	[dlrmqptype:FunctionalityQualityParameter]	DLRM Quality Parameter Type Vocabulary
Interoperability Support	[dlrmqptype:InteroperabilitySupport]	Interoperability Support	The Generic Quality Parameter reflecting the capability of a Digital Library to inter-operate	Class	[dlrmqptype:GenericQualityParameter]	DLRM Quality Parameter Type Vocabulary

Term	URI	Label	Definition	Type of term	Subclass of	Member Of
			with other Digital Libraries			
Orthogonality	[dlrmqptype:Orthogonality]	Orthogonality	The Functionality Quality Parameter which indicates to what extent different Functions are independent from each other, i.e. do not affect each other.	Class	[dlrmqptype:FunctionalityQualityParameter]	DLRM Quality Parameter Type Vocabulary
Performance	[dlrmqptype:Performance]	Performance	The Generic Quality Parameter measuring the accomplishments of a Resource	Class	[dlrmqptype:GenericQualityParameter]	DLRM Quality Parameter Type Vocabulary
PolicyQualityParameter	[dlrmqptype:PolicyQualityParameter]	Policy Quality Parameter	A Quality Parameter that concerns an aspect of the top-level Policy concept	Class	[dlrm:QualityParameter]	DLRM Quality Parameter Type Vocabulary
Reputation	[dlrmqptype:Reputation]	Reputation	The Generic Quality Parameter which reflects the trustworthiness of a Digital Library.	Class	[dlrmqptype:GenericQualityParameter]	DLRM Quality Parameter Type Vocabulary
Robustness	[dlrmqptype:Robustness]	Robustness	The Functionality Quality Parameter measuring the resilience to ill-formed input or incorrect invocation sequences of a Function.	Class	[dlrmqptype:FunctionalityQualityParameter]	DLRM Quality Parameter Type Vocabulary

Term	URI	Label	Definition	Type of term	Subclass of	Member Of
Scalability	[dlrmqptype:Scalability]	Scalability	The Generic Quality Parameter measuring the capability to increase Capacity as much as needed	Class	[dlrmqptype:GenericQualityParameter]	DLRM Quality Parameter Type Vocabulary
SecurityEnforcement	[dlrmqptype:SecurityEnforcement]	Security Enforcement	The Generic Quality Parameter which reflects the level and kind of security features offered by a Digital Library	Class	[dlrmqptype:GenericQualityParameter]	DLRM Quality Parameter Type Vocabulary
Sustainability	[dlrmqptype:Sustainability]	Sustainability	The Generic Quality Parameter which reflects the prospects of lastingness and future development of a Digital Library	Class	[dlrmqptype:GenericQualityParameter]	DLRM Quality Parameter Type Vocabulary
Usability	[dlrmqptype:Usability]	Usability	The Functionality Quality Parameter which indicates the ease of use of a given Function.	Class	[dlrmqptype:FunctionalityQualityParameter]	DLRM Quality Parameter Type Vocabulary
UserQualityParameter	[dlrmqptype:UserQualityParameter]	User Quality Parameter	A Quality Parameter that concerns an aspect of the User Domain main concept (of the Actors)	Class	[dlrm:QualityParameter]	DLRM Quality Parameter Type Vocabulary
UserSatisfaction	[dlrmqptype:UserSatisfaction]	User Satisfaction	The Functionality Quality Parameter which indicates how much an Actor is satisfied by a given Function.	Class	[dlrmqptype:FunctionalityQualityParameter]	DLRM Quality Parameter Type Vocabulary

DLRM Type Vocabulary

The DLRM Type Vocabulary contains terms denoting types of resources managed by a digital library, which are not included in the DCMI Type Vocabulary. In the context of describing digital libraries, any resource managed by the digital library, as well as the digital library as a whole, may be the “described resource” (Dublin Core Metadata Initiative, 2007). Hence we need the categories listed below to cover the full range of resources that may need to be described. The DLRM Vocabulary is meant to provide a supplement to – and be used together with - the DCMI Type Vocabulary.

Term	URI	Label	Definition	Type of term	Subclass of	MemberOf
Actor	[dcterms:Agent]	Actor	A Resource that represents an external entity that interacts with the Digital Library . It may have at an Actor Profile and belong to at a Group and be regulated by a set of Policies. An Actor may be characterized by Quality Parameters and may be linked to other Actors.	Class	[dlrm:Resource]	[dlrm:Resource]
DigitalLibrary	[dlrm:DigitalLibrary]	Digital Library	An organisation, which might be virtual, that comprehensively collects, manages and preserves f Information Objects, and offers to its Actors specialised Functions on those Information Objects, of measurable quality, expressed by Quality Parameters, and according to codified Policies.	Class	[rdfs:Resource]	DLRM Type Vocabulary

Term	URI	Label	Definition	Type of term	Subclass of	MemberOf
Function	[dlrm:Function]	Function	An operation that can be realized on a Resource or Resource Set as the result of the activity of a particular Actor. It may be performed by the Actor or it may refer to the respective supporting process of the DLS	Class	[dlrm:Resource]	DLRM Type Vocabulary
InformationObject	[dlrm: InformationObject]	Information Object	The main Resource of the Content Domain. It may have Metadata, Annotations and multiple Editions, Views, Manifestations. In addition, it may have Quality Parameters and Policies.	Class	[dlrm:Resource]	DLRM Type Vocabulary
Measure	[dlrm:Measure]	Measure	A process for computing and assigning a value to a Quality Parameter according to a unit of measurement.	Class	[dlrm:Resource]	DLRM Type Vocabulary
olicy	[dlrm:Policy]	Policy	A condition, rule, term or regulation governing the operation of a Digital Library.	Class	[dlrm:Resource]	DLRM Type Vocabulary

Term	URI	Label	Definition	Type of term	Subclass of	MemberOf
Quality Parameter	[dlrm:QualityParameter]	Quality Parameter	A Resource that indicates, or is linked to, performance or fulfilment of requirements by another Resource. A Quality Parameter is evaluated by (<evaluatedBy>) a Measure, is <measuredBy> a Measurement, and expresses the assessment (<expressAssessment>) of an Actor	Class	[dlrm:Resource]	DLRM Type Vocabulary
ResourceFormat	[dlrm:ResourceFormat]	Resource Format	A description of the structure of a Resource. May build explicitly on an Ontology or imply an Ontology.	Class	[rdfs:Resource]	DLRM Type Vocabulary

APPENDIX 4 - Property summary

Describing any objects

Poperty	URI	Label	Super-property of	Sub-property of	Domain within DLRM Voc.	Range within DLRM Voc.
description	[dcterms:description]	Description			[rdfs:Resource]	[rdfs:Resource]
title	[dcterms:title]	Title			[rdfs:Resource]	[rdfs:Resource]
type	[dcterms:type]	Type			[rdfs:Resource]	[rdfs:Class]

Describing digital libraries

Poperty	URI	Label	Super-property of	Sub-property of	Domain within DLRM Voc.	Range within DLRM Voc.
agreeWith	[dlrm:agreeWith]	Agree With			[dlrm:DigitalLibrary]	[dcterms:Policy]
isSupportedBy	[dlrm:support] [dlrm:isSupportedBy]	is Supported By			[dlrm:DigitalLibrary]	[dlrm:DigitalLibrarySystem]
manage	[dlrm:manage]	Manage			[dlrm:DigitalLibrary]	[dlrm:Resource]
offer	[dlrm:offer]	Offer			[dlrm:DigitalLibrary]	[dlrm:Function]

Poperty	URI	Label	Super-property of	Sub-property of	Domain within DLRM Voc.	Range within DLRM Voc.
serve	[dlrm:serve]	Serve			[dlrm:DigitalLibrary]	[dcterm:Agent]
tender	[dlrm:tender]	Tender			[dlrm:DigitalLibrary]	[dlrm:QualityParameter]

Describing resources managed by the digtal libraries

Resources in general

Poperty	URI	Label	Super-property of	Sub-property of	Domain within DLRM Voc.	Range within DLRM Voc.
associatedWith	[dlrm:associatedWith]	Associated With		[dcterms:relation]	[dlrm:Resource]	[dlrm:Resource]
belongTo	[dlrm:belongTo]	Belong To		[dcterms:relation]	[dlrm:Resource]	[dlrm:ResourceSet]
describedBy	[dlrm:describedBy]	Described By		[dlrm:hasMetadata]	[dlrm:Resource]	[dlrm:InformationObject]
expressedBy	[dlrm:expressedBy]	Expressed By		[dcterms:relation]	[dlrm:Resource]	[dlrm:Information Object]
hasAnnotation	[dlrm:hasAnnotation]	Has Annotation		[dcterms:relation]	[dlrm:Resource]	[dlrm:InformationObject] (Annotation)
hasFormat	[dlrm:hasFormat]	Has Format	[dcterms: format]		[dlrm:Resource]	[dlrm:ResourceFormat]

Poperty	URI	Label	Super-property of	Sub-property of	Domain within DLRM Voc.	Range within DLRM Voc.
hasMetadata	[dlrm:hasMetadata]	Has Metadata	[dlrm: describedBy]	[dcterms:relation]	[dlrm:Resource]	[dlrm:InformationObject] (Metadata)
hasPart	[dcterms:hasPart]	Has Part			[dcterms: relation]	[dlrm:Resource]
hasPurpose	[dlrm:hasPurpose]	Has Purpose			[dlrm:Resource]	[dlrm:Purpose]
hasQuality	[dlrm:hasQuality]	Has Quality		[dcterms:relation]	[dlrm:Resource]	[dlrm:QualityParameter]
identifiedBy	[dlrm:identifiedBy]	Identified By			[dlrm:Resource]	[dlrm:ResourceIdentifier]
regulatedBy	[dlrm:regulatedBy]	Regulated By	[dcterms: rights]	[dcterms:relation]	[dlrm:Resource]	[dcterms:Policy]

Functions

Poperty	URI	Label	Super-property of	Sub-property of	Domain within DLRM Voc.	Range within DLRM Voc.
actOn	[dlrm:actOn]	Act On			[dlrm:Function]	[dlrm:Resource]
create	[dlrm:create]	Create		[dcterms:relation]	[dlrm:Function]	[dlrm:Resource]

Poperty	URI	Label	Super-property of	Sub-property of	Domain within DLRM Voc.	Range within DLRM Voc.
functionType	[dlrm:functionType]	Type Of Function			[dlrm:Function]	[rdfs:Class]
influencedBy	[dlrm:influencedBy]	Influenced By		[dcterms:relation]	[dlrm:Function]	[dlrm:InformationObject] (ActorProfiles)
interactWith	[dlrm:interactWith]	Interact With		[dcterms:relation]	[dlrm:Function]	[dlrm:Function]
retrieve	[dlrm:retrieve]	Retrieve		[dcterms:relation]	[dlrm:Function] (AccessResource)	[dlrm:Resource]
return	[dlrm:return]	Return		[dlrm:retrieve]	[dlrm:Function] (Discover)	[dlrm:ResourceSet] (ResultSet)
issue	[dlrm:issue]	Issue		[dcterms:relation]	[dlrm:Function] (Search)	[dlrm:InformationObject] (Query)

Queries

Poperty	URI	Label	Super-property of	Sub-property of	Domain within DLRM Voc.	Range within DLRM Voc.
produce	[dlrm:produce]	produce		[dcterm:relation]	[dlrm:InformationObject] (Query)	[dlrm:ResourceSet] (ResultSet)

Policies

Poperty	URI	Label	Super-property of	Sub-property of	Domain within DLRM Voc.	Range within DLRM Voc.
govern	[dlrm:govern]	Govern		[dcterms:relation]	[dlrm:Policy]	[dlrm:Resource]
grantedTo	[dlrm:grantedTo]	Granted To		[dcterms:relation]	[dlrm:Policy]	[dcterms:Agent]
isEnforced	[dlrm:isEnforced]	Is Enforced			[dlrm:Policy]	[rdfs:#Literal]
isExplicit	[dlrm:isExplicit]	Is Explicit			[dlrm:Policy]	[rdfs:#Literal]
isExtrinsic	[dlrm:isIntrinsic]	Is Intrinsic			[dlrm:Policy]	[rdfs:#Literal]
isPrescriptive	[dlrm:isPrescriptive]	Is Prescriptive			[dlrm:Policy]	[rdfs:#Literal]
policyType	[dlrm:policyType]	Type Of Policy			[dlrm:POlicy]	[rdfs:Class]

Quality Parameters

Poperty	URI	Label	Super-property of	Sub-property of	Domain within DLRM Voc.	Range within DLRM Voc.
evaluatedBy	[dlrm:evaluatedBy]	Evaluated By			[dlrm:Quality Parameter]	[dlrm:Measure]

Poperty	URI	Label	Super-property of	Sub-property of	Domain within DLRM Voc.	Range within DLRM Voc.
measuredBy	[dlrm:measuredBy]	Measured By			[dlrm:Quality Parameter]	[dlrm:Measurement] (in test: a Literal value)
expressAssessment	[dlrm:expressAssessment]	Express Assessment		[dcterms:relation]	[dlrm:Quality Parameter]	[dcTerms:Agent]
affectedBy	[dlrm:affectedBy]	Affected By		[dcterms:relation]	[dlrm:QualityParameter]	[dlrm:Resource]
qualityParameterType	[dlrm:qualityParameterType]	Type Of Quality Parameter			[dlrm:QualityParameter]	[rdfs:Class]

Measure

Poperty	URI	Label	Super-property of	Sub-property of	Domain within DLRM Voc.	Range within DLRM Voc.
measureType	[dlrm:measureType]	Type of Measure			[dlrm:Measure]	[rdfs:Class]