

Digital Library Interoperability at High Level of Abstraction

Maristella Agosti, Nicola Ferro*, Gianmaria Silvello

Department of Information Engineering, University of Padua, Via Gradenigo 6/a, 35131 Padua, Italy

Abstract

Digital Library (DL) are the main conduits for accessing our cultural heritage and they have to address the requirements and needs of very diverse memory institutions, namely *Libraries, Archives and Museums (LAM)*. Therefore, the interoperability among the *Digital Library System (DLS)* which manage the digital resources of these institutions is a key concern in the field.

DLS are rooted in two foundational models of what a digital library is and how it should work, namely the *DELOS Reference Model* and the *Streams, Structures, Spaces, Scenarios, Societies (5S)* model. Unfortunately these two models are not exploited enough to improve interoperability among systems.

To this end, we express these foundational models by means of ontologies which exploit the methods and technologies of Semantic Web and Linked Data. Moreover, we link the proposed ontologies for the foundational models to those currently used for publishing cultural heritage data in order to maximize interoperability.

We design an ontology which allows us to model and map the high level concepts of both the 5S model and the DELOS Reference Model. We provide detailed ontologies for all the domains of such models, namely the user, content, functionality, quality, policy and architectural component domains in order to make available a working tool for making DLS interoperate together at a high level of abstraction. Finally, we provide a concrete use case about digital annotation of illuminated manuscripts to show how to apply the proposed ontologies and illustrate the achieved interoperability between the 5S and DELOS Reference models.

Keywords: digital library, digital library system, digital library foundational models, 5S model, DELOS Reference Model, interoperability, digital library interoperability

1. Introduction

Digital Library (DL) have been steadily progressing since the early 1990s and they now determine how citizens and organizations study, learn, access and interact with their cultural heritage [1–8]. Despite their name, DL are not only the digital counter-part of traditional libraries but they are also concerned with other kinds of cultural heritage institutions, such as archives and museums, that is institutions typically referred to as *Libraries, Archives and Museums (LAM)*. In the context of LAM, unifying a variety of organizational settings and providing more integrated access to their contents are aspects of utmost importance. Although the type of materials may differ and professional practices vary, LAM share an overlapping set of functions and fulfilling them in “collaboration rather than isolation creates a win-win for users and institutions” [9].

These compelling integration and collaboration needs have propelled the evolution of *Digital Library System (DLS)* [10] as systems that permit us to design and implement the overlapping set of functions of LAM.

In the 1990s, DLS were monolithic systems, each one built for a specific kind of information resource – e.g. text, images, or

videos – and with very specialised functionalities developed ad-hoc for those contents and their reference users. This approach caused a flourishing of systems where the very same functionalities, e.g. user management or repositories, were developed and re-developed from scratch many times, causing them to be different and often incompatible one with each other. From the mid 2000s, DLS evolved towards service-oriented architectures, where components can be plugged into each other to provide the desired end-user functionalities, yet requiring careful and ad-hoc configuration. This paradigm shift allowed DLS to become more and more user-centered systems, where the original content management functionality was partnered with new communication and cooperation functionalities such as user annotation [11], with the ultimate goal of acting as “a common vehicle by which everyone can access, discuss, evaluate and enhance information of all forms” [12]. As a consequence, DLS started to embody the above vision for LAM since they were no longer isolated systems but, on the contrary, they needed to cooperate together in order to improve the user experience in accessing information and to seamlessly integrate information resources of different cultural heritage institutions.

This evolution has been favored by the development of two foundational models of what DL are, namely the *Streams, Structures, Spaces, Scenarios, Societies (5S)* model [13] and the *DELOS Reference Model* [14], which made it clear what kind of entities should be involved in a DL, what their functionalities

*corresponding author

Email addresses: maristella.agosti@unipd.it (Maristella Agosti), nicola.ferro@unipd.it (Nicola Ferro), gianmaria.silvello@unipd.it (Gianmaria Silvello)

should be and how DLS components should behave, and fostered the design and development of operational DLS complying with them.

However, these two models are quite abstract and, still providing a unifying vision of what a DL is, they allow for very different choices when it comes to develop actual DLS. This has led to the growth of “ecosystems” where services and components may be able, at best, to interoperate together within the boundaries of DLS that have been inspired by just one of the two models for DL. However, there are no running examples of two DLS, one implementing the 5S model and the other the DELOS Reference Model, which are able to interoperate. Therefore, interoperability still represents one of the biggest challenges in the DL field [10, 15].

In this work, we address a still open issue in the DL realm: to make DL foundational models interoperable in order to derive all the other kinds of interoperability, in particular, interoperability between operational DLS. The main contributions of the paper are:

- a detailed analysis of the 5S and DELOS Reference models pointing out common aspects and main differences;
- the definition of a common ontology which encompasses and links the concepts of the DELOS Reference Model and the 5S models, covering all the domains of such models: the user, content, functionality, quality, policy and architectural component;
- a concrete use case about digital annotation of illuminated manuscripts to show how to apply the proposed ontologies and illustrate the achieved interoperability between the 5S and DELOS Reference models.

The paper is organized as follows: Section 2 illustrates the rationale of the paper; Section 3 reports on some pertinent related works; Section 4 introduces the relevant aspects of the 5S Model and of the DELOS Reference Model; Section 5 presents the semantic mapping between the 5S Model and the DELOS Reference Model associating the high level concepts of one model to those of the other; Section 6 to 11 respectively show the correspondences between the notions and domains of user, content, functionality, quality, policy and architectural component in the two models; Section 12 presents a relevant case of application of the proposed approach where users who interact with two DLS, that manage illuminated manuscripts, are interested in annotating their contents to perform activities of their interest. Section 13 sums up the results presented in the paper.

2. Rationale

The current mainstream approach to bridge the interoperability gap between DLS and to provide comprehensive solutions able to embrace the full spectrum of LAM is to exploit semantic Web technologies and linked (open) data [16, 17]. This allows for describing entities and information resources in a common way which enables their exchange, as for example happens in the case of library linked data [18].

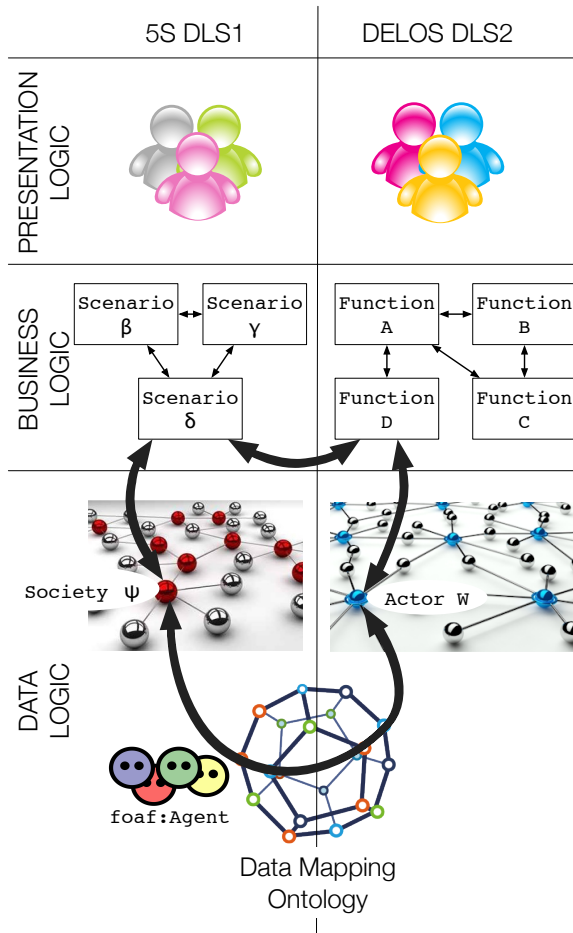
This approach is both “external” and “bottom-up”. It is “external” since it assumes that everything in a DL should be exposed on the Web rather than seeking direct interoperability among systems which may not necessarily be only Web-based. It is “bottom-up” because ontologies have been used only to describe the resources managed by a DLS and they are not used to represent the concepts themselves which constitute the DL model on which the DLS is based. Therefore, they allow for semantic interoperability and integration only at the data level, i.e. the lowest level possible in the architecture of a DLS. Indeed, at present different operational DLS, even if they are based on the same DL model, may not fully interoperate because the mapping between the abstract foundational model, either the DELOS or the 5S one, and the actual one implemented by the DLS is not explicitly provided. As a consequence, each operational DLS may not actually “know” the way in which another operational DLS calls and refers to the same operations.

Consider Figure 1(a) which depicts the current approach. If we need two operations in two different DLS to interoperate, we need to create some kind of link between them at the business logic level. This link is typically manual and hard-coded, e.g. by direct invocation of the respective functionalities, since each DLS has no knowledge or understanding of each others internals. Each operation in each DLS makes use and processes some kind of data at the data logic level. Usually, these data need to be mapped to some common format, through some external ontology, to be exchanged between the two systems, due to lack of reciprocal knowledge. Because of that, we consider this as a “bottom-up” approach since the data level, the one where interoperability is achieved, is the lowest level in the architecture of a DLS.

As a concrete example of what is depicted in Figure 1(a), suppose that two different operational DLS, one built using the 5S model and the other built using the DELOS Reference Model, need to enrich a user profile by exploiting their own specific service. To this end, they both need to exchange user data and to access their specific enrichment functionalities. Since a common view of user is lacking among the two DLS, we could use the class *Agent* in *Friend of a Friend (FOAF)*¹ to represent the notion of user of a DL in order to allow the two systems to exchange the profiles. Nevertheless, the *Agent* class is neither related to the concept of *Society* nor to the concept of *Actor* which represent users in the 5S model and in the DELOS Reference Model, respectively. Therefore, to exchange the user profiles, the two different operational DLS would need a set of (hard-coded) rules instructing them how to translate the *Society* ψ and the *Actor* W into the *Agent* class. Moreover, the *Agent* class is neither related to the notion of *Scenario* in the 5S model nor to the notion of *Function* in the DELOS Reference Model, whose specializations define the services for profile enrichment. Moreover, there is no ontology which tells the operational DLS based on the 5S model that its own *Scenario* δ for profile enrichment corresponds to a specific *Function* D for profile enrichment in the DELOS Reference Model. Therefore, to enrich user profiles with reciprocal information, the two

¹<http://www.foaf-project.org/>

(a) Current Approach



(b) Proposed Approach

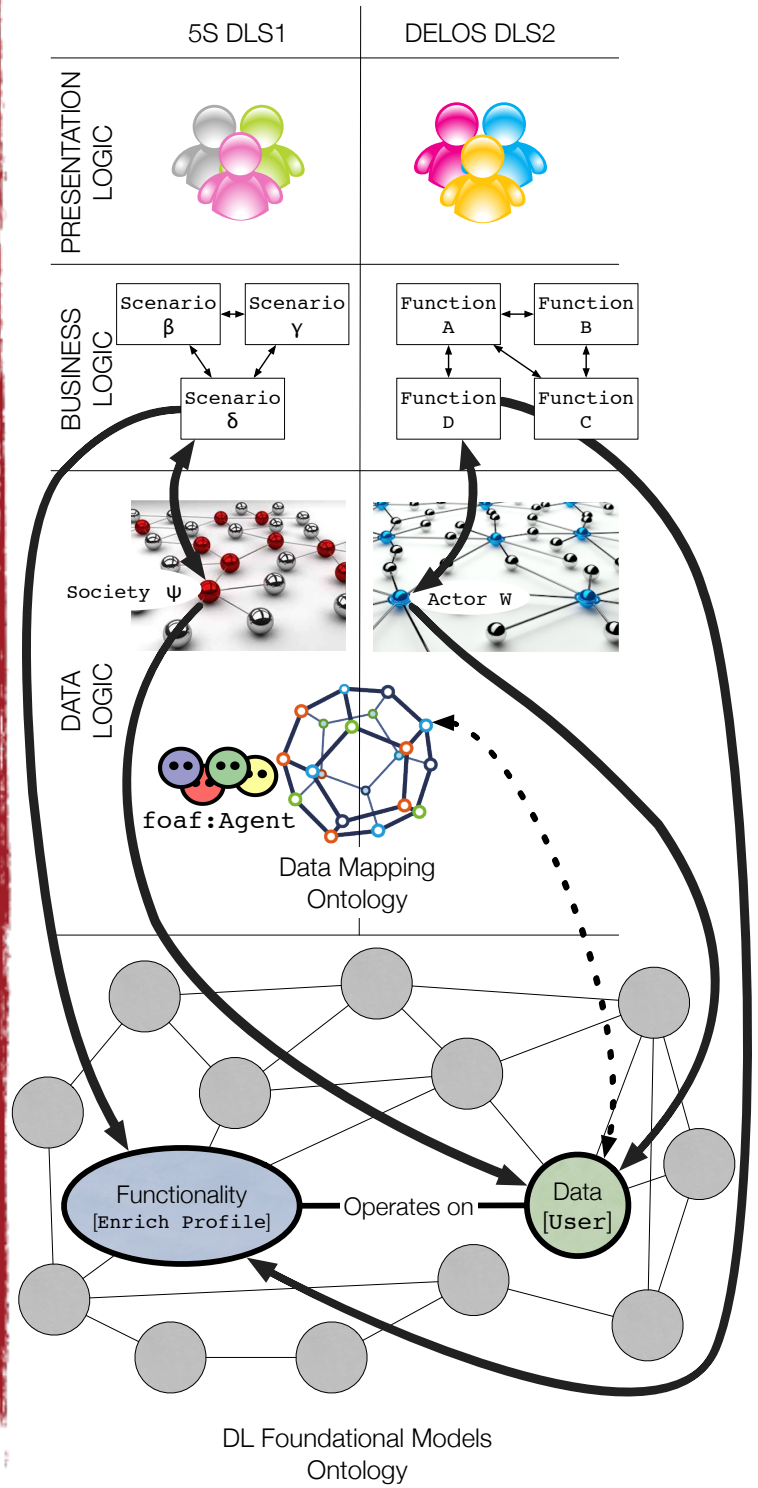


Figure 1: Current and proposed approach for DL interoperability.

different DLS would need to somehow manually (hard) code direct invocations to their respective services, also instructing them to map their internal representation of user profiles into FOAF before exchanging it. Overall, this solution would somehow allow for transferring the data among systems and invoking the respective services, but it would lose the notion of what is happening and hamper a deeper kind of interoperability among operational DLS.

What is needed is a deeper and more abstract interoperability based on a commonly shared semantic view of what a DL is rather than a lower level one where data is just wrapped in a commonly understandable format.

As shown in Figure 1(b), the quite ambitious goal of this paper is to propose a solution to this open problem. The proposed solution is based on the representation of each foundational DL model through ontologies, leveraging semantic Web and linked data technologies in order to ease their linking to other already existing ontologies and to achieve maximum interoperability. According to our proposed approach, the “DL Foundational Models Ontology”, represented in the lower part of Figure 1(b), expresses each core concept in the DL domain as well as their relationships. So, for example, we can express the fact that a given *Functionality*, in our case *Enrich Profile Operates on some Data*, in our case *User*. This happens at the “DL Foundational Models Ontology” level where functions, data and relationships are thought of as classes. We can link these classes to the corresponding instances in the different DLS and know that *Scenario δ* in DLS1 and *Function D* in DLS2 are instances of the same functionality *Enrich Profile* and that *Society ψ* in DLS1 and *Actor W* in DLS2 are instances of the same kind of data *User*. As a consequence, the two DLS would “know”, by passing through the “DL Foundational Models Ontology”, which are the corresponding operations they want to use, and they would also “know” the meaning of the data associated to those operations and, if necessary, they could decide to map them through an external data mapping ontology.

Therefore, the proposed approach will pave the way for a deeper interoperability among operational DLS and lower the barriers between LAM. It is also opening up more advanced possibilities for the automatic processing of resources, since, for example, DLS could automatically exploit the link between the models they are built upon in order to exchange resources, interoperate and integrate functionalities.

Note that Figure 1 is intentionally abstract and not bound to any specific technology or architectural paradigm. Indeed, even if there are many ways to instantiate an operational DLS, e.g. by using Web services, the issues just discussed do not primarily derive from any of them but rather from the lack of a shared view, processable in an automatic way, of the DL foundational concepts. Once this view is built, as we propose in this paper, several technologies are suitable to instantiate and implement it.

3. Related Work

As anticipated in the previous sections, most of the work that has been conducted to address the problem of interoperabil-

ity in DL is related to the representation and exchange of data managed by a DLS, typically via *Linked Open Data (LOD)* [17], i.e. it is mostly concerned with the data logic, as represented in Figure 1. This explains the need to develop many ontologies and metadata schemes for this purpose, and the number of ontologies and metadata schemes is continuously growing along with the number of specific cases.

When it comes to ontologies, we have, for example, actor ontologies – e.g. FOAF² or BIO³; place ontologies – e.g. GeoNames⁴; time ontologies – e.g. the time period encoding scheme⁵ in the Dublin Core Metadata Terms; event ontologies – e.g. the Event Ontology⁶ or *Linking Open Descriptions of Events (LODE)*⁷; and many many others.

When it comes to metadata schemes, we have different options for different cultural heritage institutions: for example, *Metadata Object Description Schema (MODS)*⁸, Dublin Core [19], and *Metadata Authority Description Standard (MADS)*⁹ for libraries; *Encoded Archival Description (EAD)* [20] and *Encoded Archival Context for Corporate Bodies, Persons, and Families (EAC-CPF)*¹⁰ for archives; *Lightweight Information Describing Objects (LIDO)* [21] for museums.

There are also ontologies and metadata schemes designed for favoring the exchange of information, such as *Open Archives Initiative Object Reuse and Exchange (OAI-ORE)* [22], *CIDOC Conceptual Reference Model (CIDOC CRM)* [23], or *Metadata Encoding and Transmission Standard (METS)*¹¹. As a further example, we can consider the *Europeana Data Model (EDM)* [24–26], which aims at structuring the data managed by Europeana¹², a major effort of the European Union to create a DL containing the cultural heritage of Europe. A common model like EDM can instead be seen as an anchor to which various finer-grained models can be connected, making them at least partly interoperable at the semantic level, while the data retain their original expressivity and richness.

All the above described solutions are certainly key blocks for enabling interoperability but are all, in one way or another, confined to what in Figure 1 is labelled “Data Mapping Ontology”.

When it comes to the business logic layer, there are both ad-hoc solutions, such as *Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH)* [27] which is specifically focused on exchanging metadata among DLS, and general solutions, such as Web Services [28, 29] even enriched with semantic annotations [30]. While OAI-PMH is basically just a transport mechanism, Web Services can clearly play a role in allowing for functional interoperability among DLS. However,

²<http://www.foaf-project.org/>

³<http://vocab.org/bio/0.1/.html>

⁴<http://geonames.org/>

⁵<http://dublincore.org/documents/dcmi-period/>

⁶<http://motools.sourceforge.net/event/event.html>

⁷<http://linkedevents.org/ontology/>

⁸<http://www.loc.gov/standards/mods/>

⁹<http://www.loc.gov/standards/mads/>

¹⁰<http://eac.staatsbibliothek-berlin.de/>

¹¹<http://www.loc.gov/standards/mets/>

¹²<http://www.europeana.eu/>

they can be considered “external”, in the sense discussed in the previous section since they would assume a DLS is exposed on the Web, which may not always be the case. However, even in the case we want to exploit Web Services enriched with semantic annotations, an ontology for annotating the Web Services is needed and this common ontology is lacking today in the DL domain and it would be a part of what we called “DL Foundational Models Ontology” in the previous section.

Therefore, to the best of our knowledge, there is no previous work in the field which attempted to achieve interoperability among DLS at a high level of abstraction through a semantic description and mapping of their foundational models. We can only mention our very preliminary work [31], where we started to explore this idea in the context of quality in DL.

4. Foundational Models for Digital Libraries

The 5S [8, 13, 32] is a formal model which draws upon the broad digital library literature to produce a comprehensive base of support. It was developed largely bottom up, starting with key definitions and elucidation of digital library concepts from a minimalist approach.

The DELOS Reference Model [14] is a high-level conceptual framework that aims at capturing significant entities and their relationships within the digital library universe with the goal of developing a more robust model of it.

The DELOS Reference Model and the 5S model address a similar problem with different approaches. The 5S is a formal model providing mathematical definitions of the digital library entities that can be used to prove properties, theorems and propositions. The DELOS Reference Model does not provide formal definitions, but it does provide a way to model and manage the resources of the digital library realm by using concept maps [33, 34] because of their simplicity and immediacy.

4.1. 5S Model

The 5S is a formal model for digital libraries based on the following abstractions:

- *Streams* are sequences of elements of an arbitrary type (e.g. bits, characters, images) and thus they can model both static and dynamic content. Static streams correspond to information content represented as basic elements, e.g. a simple text is a sequence of characters, while a complex object like a book may be a stream of simple text and images. Dynamic streams are used to model any information flow and thus are important for representing any communication that takes place in the digital library. Finally, streams are typed and the type is used to define their semantics and application area.
- *Structures* are the way through which parts of a whole are organised. In particular, they can be used to represent hypertexts and structured information objects, taxonomies, system connections and user relationships.
- *Spaces* are sets of objects together with operations on those objects conforming to certain constraints. Document spaces are the key concepts in digital libraries. However, spaces are used in various contexts, e.g. indexing and visualising. Different types of spaces are proposed, e.g. measurable, measure, probability, vector and topological spaces.
- *Scenarios* are sequences of events that may have parameters, and events that represent state transitions. Thus a scenario tells what happens to the streams in spaces and through the structures. When considered together, the scenarios describe the services, the activities and the tasks representing digital library functions. Work-flows and data-flows are examples of scenarios.
- *Societies* are sets of entities and relationships. The entities may be humans or software and hardware components, which either use or support digital library services. Thus, society represents the highest-level concept of a digital library, which exists to serve the information needs of its societies and to describe the context of its use.

We can relate the 5S model to some of the aims of a digital library: *societies* define how a digital library helps in satisfying the information needs of its users; *scenarios* provide support for the definition and design of different kinds of services; *structures* support the organisation of the information in usable and meaningful ways; *spaces* deal with the presentation and access to information in usable and effective ways; *streams* concern the communication and consumption of information by users.

Building upon the five main concepts, the model provides a definition for a *minimal digital library*, as depicted in Figure 2, and it is constituted by: (i) a repository of digital objects; (ii) a set of metadata catalogues containing metadata specifications for those digital objects; (iii) a set of services containing at least services for indexing, searching, and browsing; and, (iv) a society.

Starting from the definition of a minimal digital library, [35] developed a set of tools aimed at automatically instantiating and deploying a DLS from a catalog of components corresponding to the notions introduced in the model. Unfortunately, this vision has not been fully embodied yet, especially in wide settings, but the last decade of efforts geared towards interoperability and the pervasiveness today of semantic Web technologies may offer the opportunity of performing the next step in this direction.

As far as quality is concerned, a separate quality model has been developed [36]. For each major digital library concept in the 5S model, a number of *Quality Dimensions* are formally defined and a set of *Numerical Indicators* for those quality dimensions are proposed. In particular, they consider key concepts of a minimal digital library: Digital Object, Metadata Specification, Collection, Metadata Catalogue, Repository and Services. For some key concepts, pairs of form (quality dimension, numerical indicator) are illustrated through their application to a number of “real-world” digital libraries. To help operationalize

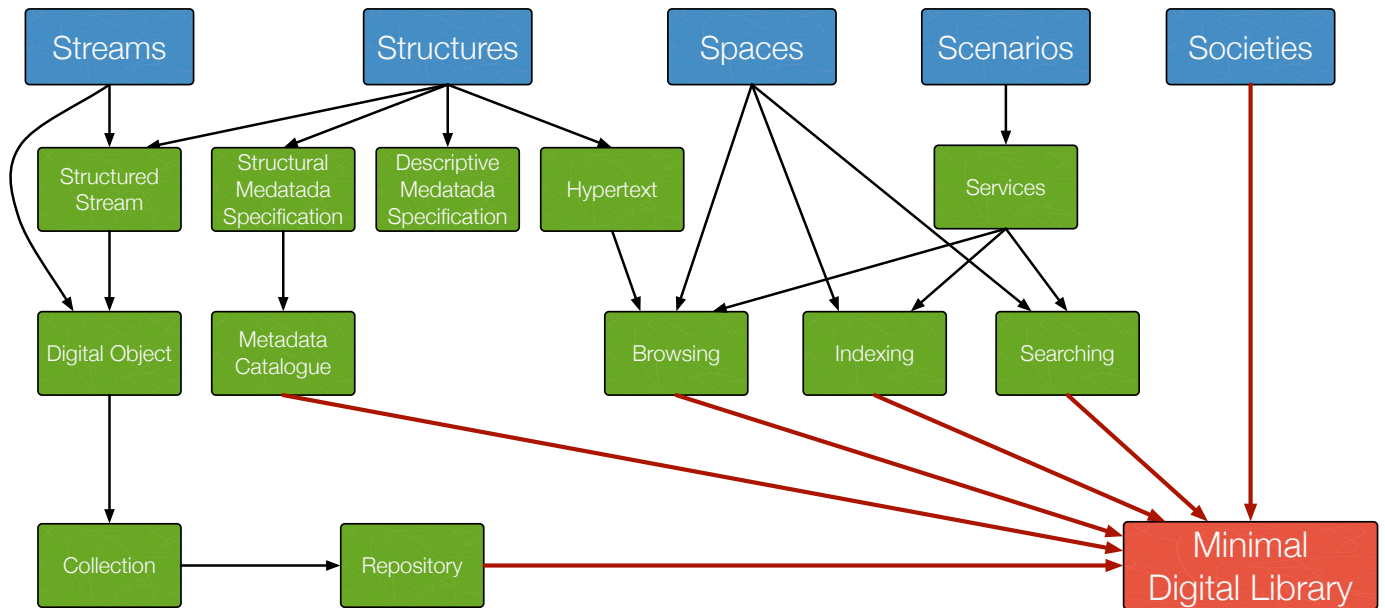


Figure 2: Definition of a minimal digital library according to the 5S model.

this approach, a digital library quality assessment toolkit has been developed and deployed [37]. It can be used by digital library managers to assess the quality of their digital library, based on the toolkits processing of system logs and its access to digital library content.

4.2. DELOS Reference Model

The DELOS Reference Model approaches the problem of modeling the digital library universe by highlighting six domains or main concepts, as shown in Figure 3(a), which lie at the core of what digital libraries are and what their purposes are:

- **Content:** the data and information that digital libraries handle and make available to their users.
- **User:** the actors (whether human or not) entitled to interact with digital libraries.
- **Functionality:** the services that digital libraries offer to their users.
- **Quality:** the parameters that can be used to characterize and evaluate the content and behaviour of digital libraries.
- **Policy:** a set of rules that govern the interaction between users and digital libraries.
- **Architecture:** a mapping of the functionalities and content offered by a digital library into hardware and software components.

These six domains represent the high level containers that help to structure the DELOS Reference Model. For each of these concepts, the fundamental entities and their relationships

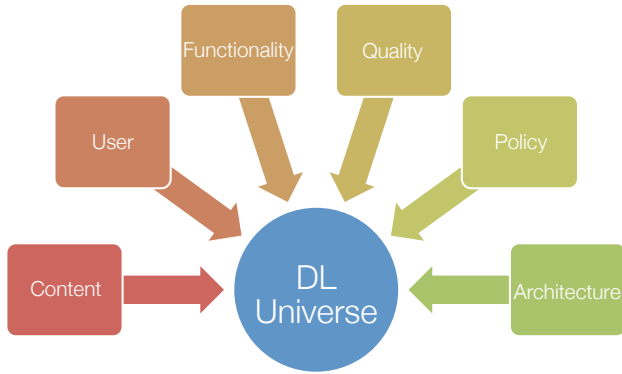
are clearly defined and discussed. Note that these six domains are not separate, but, on the contrary, are strongly inter-related; the entities within a domain are often related to or influenced by the entities in other domains.

Besides, the DELOS Reference Model distinguishes between three different “systems” which constitute the digital library universe, as shown in Figure 3(b), and rely on the six domains introduced above for their definition:

- **Digital Library (DL):** an organisation, which might be virtual, that comprehensively collects, manages and preserves for the long term rich *digital contents*, and offers to its *user* communities with specialised *functionalities* on that content, of measurable *quality* and according to codified *policies*.
- **Digital Library System (DLS):** a software system based on a defined (possibly distributed) *architecture* that provides all *functionalities* required by a particular *Digital Library*. *Users* interact with a Digital Library through the corresponding Digital Library System.
- **Digital Library Management System (DLMS):** a generic software system that provides the appropriate software infrastructure both (i) to produce and administer a Digital Library System incorporating the suite of *functionalities* considered fundamental for Digital Libraries and (ii) to integrate additional software offering more refined, specialised or advanced *functionalities*.

The three “systems” are at different levels of abstractions and constitute a kind of hierarchy: at the more general level there is the notion of DL, which is what is actually perceived by the end-users and what they interact with; in-between, there is the DLS, which mainly concerns system designers and administrators who have to instantiate and manage it; at the lower

(a) DELOS Reference Model Domains



(b) DELOS Reference Model "Systems"

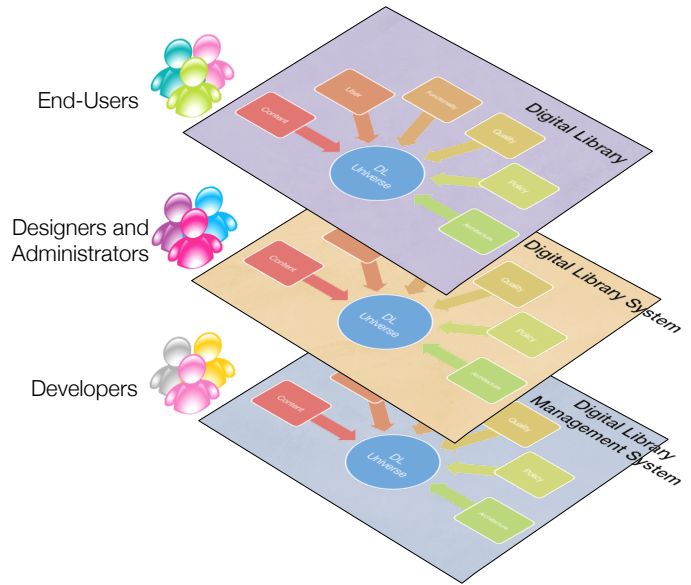


Figure 3: Main concepts of the DELOS Reference Model.

level, there is the DLMS, which typically interests system developers who implement the actual components that are used by the upper layers.

These three systems constitute a hierarchy in the sense that entities and definitions introduced at a more general level are inherited by the levels underneath and can be further specialised by them; in addition, a lower level can introduce new definitions and entities that are specific only to that level. In this way, each one of the three systems contributes in an incremental way to the modeling of each one of the six domains introduced above.

5. Semantic Mapping between the 5S Model and the DELOS Reference Model

In Figure 4 we present the *Resource Description Framework (RDF)* graph of the unifying data model relating the DELOS Reference Model to the 5S model by means of a mapping between their most relevant high-level concepts. The presented RDF graph is composed by classes represented as circles and properties represented as directed edges between the classes. In this section we report the main concepts composing the two models and their mapping one into the other, whereas in the following sections we analyze in detail each single domain of the models reporting their connections with external classes in the LOD cloud. In Table 1 we report the namespaces and prefixes of the vocabularies adopted for defining the RDF classes and properties.

The main constituents of the DELOS Reference Model are: the digital universe divided into DL, DLS and DLMS, the con-

cept of *Resource* and six high-level main domains: User, Functionality, Content, Quality, Policy and Architecture.

A DL, represented by the *Digital Library* class in Figure 4, is supported by a *Digital Library System* which is extended and deployed by a *Digital Library Management System*. These three classes manage *Resources*, where a *Resource* is any identifiable entity in the DL universe and resembles the concept of resource used in the Web [38]. A *Resource* represents the class of everything that exists in the DL universe and it is related to the `rdfs:Resource` class. In addition to this general concept, the *Resource* in the DELOS Reference Model has some additional features: it can be arranged or set out according to a resource format which, for example, allows a *Resource* to be composed of or linked to other *Resources*.

All the DELOS domains are represented by key classes which are subclasses of *Resource*.

In the user domain these key classes are *Actor* and *Group*, where a *Group* may be a collection of actors and at the same time it is a subclass of *Actor* given that a group can be considered as a single actor itself. The user domain directly interacts with the functionality and the quality domains; indeed, an *Actor* may perform some *Functions* in the DL and may be assessed by some *Quality Parameter*.

The *Function* class always acts on a resource and the *Quality Parameter* is associated to a resource as happens in the case of *Actor*, which is related to *Quality Parameters* by the `expressAssessment` property.

The main class of the content domain is *Information Object*, which represents any information managed by the DL and includes documents such as texts, images, videos, audio

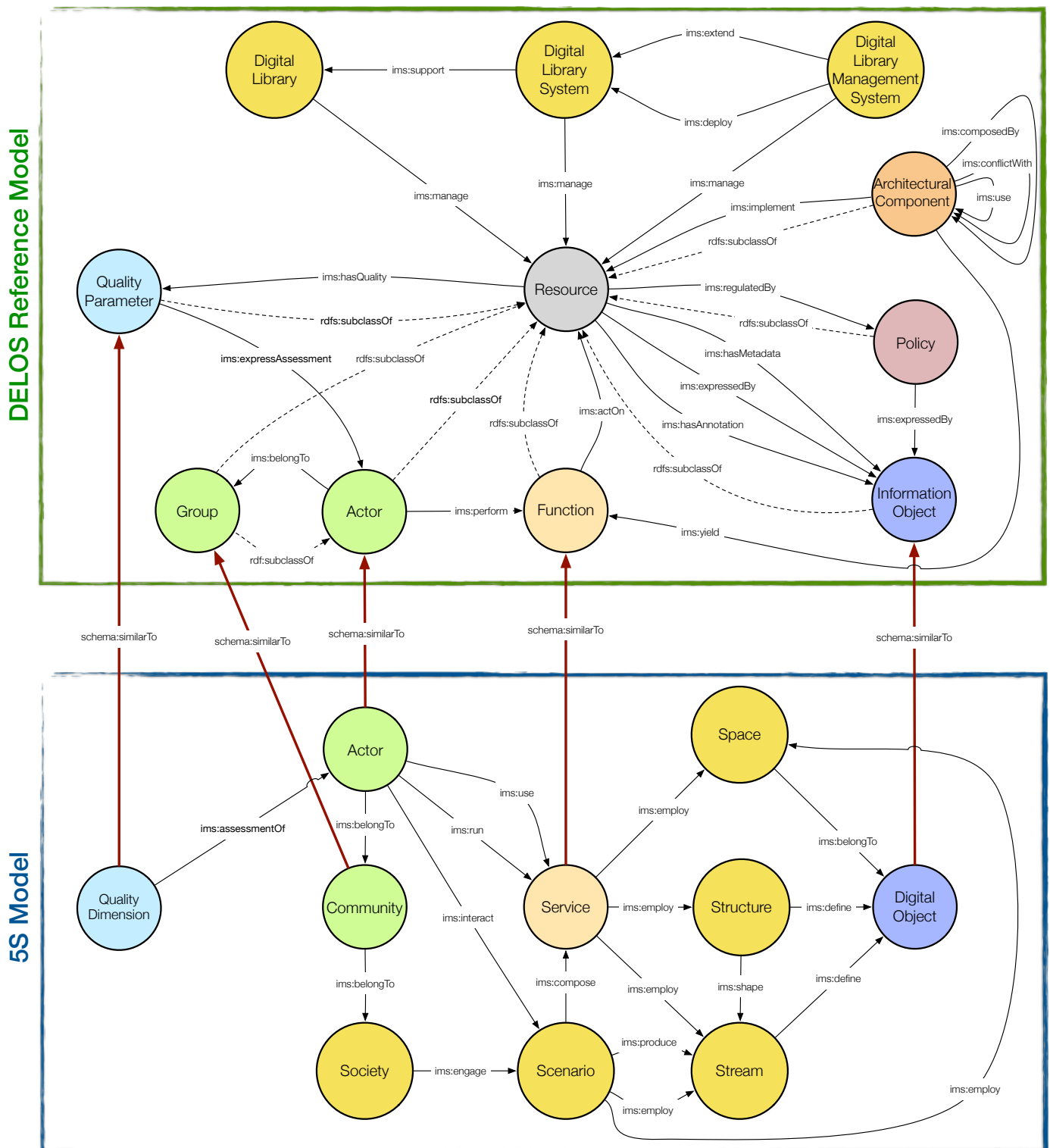


Figure 4: Semantic mapping of the high level concepts in the 5S model and DELOS Reference Model and their relationships.

Table 1: Prefixes and Namespaces of the vocabularies adopted to define the DELOS Reference Model and the 5S model. The vocabulary can be used to connect the model classes to external classes of the LOD cloud.

Prefix	Namespace	Name
cidoc	http://purl.org/NET/cidoc-crm	CIDOC-CRM Ontology
cos	http://cos.ontoware.org/cso#	Core Software Ontology
dbo	http://www.dbpedia.org/	DBpedia
dcat	http://www.w3.org/TR/vocab-dcat/	Data Catalog Vocabulary
dctypes	http://purl.org/dc/dcmitype/	Dublin Core Type Vocabulary
fabio	http://purl.org/spar/fabio/	FRBR-aligned Bibliographic Ontology
foaf	http://xmlns.com/foaf/0.1/	Friend of a friend
ims	http://ims.dei.unipd.it/data/rdf/	Information Management System (IMS) vocabulary terms
istmo	http://ontology.it/istmo/v1/istmo.html	IT Service Management Ontology
mvco	http://dmag.ac.upc.edu/ontologies/mvco/	Media Value Chain Ontology
nfo	http://www.semanticdesktop.org/ontologies/2007/03/22/nfo/	NEPOMUK File Ontology
oa	http://www.w3.org/ns/oa	Open Annotation Data Model
org	http://www.w3.org/ns/org	Core organization ontology
owl	http://www.w3.org/2002/07/owl#	OWL vocabulary terms
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#	RDF vocabulary terms
rdfs	http://www.w3.org/2000/01/rdf-schema#	RDF Schema
schema	https://schema.org/	Schema.org promotes schemas for structured data on the Internet, on web pages, in email messages, and beyond.
skos	http://www.w3.org/2009/08/skos-reference/skos.html	SKOS Simple Knowledge Organization System
snarm	http://rdf.myexperiment.org/ontologies/snarm/	Simple Network Access Rights Management Ontology
swco	http://data.semanticweb.org/ns/swc/swc_2009-05-09.html#	Semantic Web Conference Ontology

files, metadata and annotations.

The main class of the policy domain is *Policy*, which represents the single entity governing a resource with respect to a certain management point and it is connected to *Resource* by the *regulatedBy* property.

Lastly, the main class of the architecture domain is *Architectural Component*, which defines the organization or structure of the components of a given system or service; an architectural component may be composed of other components or may use other components.

The RDF model of the 5S model is represented in the lower part of Figure 4 where we can see five main classes representing the five S of the model: *Society*, *Scenario*, *Stream*, *Structure* and *Space*. A *Scenario* is engaged by a *Society* and employs or produces a *Stream* which is shaped by a *Structure*. The *Service* class is central to this model because it connects all of its fundamentals given that it employs a *Space*, a *Structure* and a *Stream*; furthermore, a *Service* is composed by one or more *Scenarios* which are related to a *Society*. A *Scenario* may employ a *Space*. The *Digital Object* class is defined by a set of streams and structures and employs some spaces.

The 5S RDF graph represents the user domain similarly to the DELOS Reference Model; indeed, the class *Actor* represents an agent (e.g. a human or a computer) which belongs to a *Community* belonging to a *Society*.

The mapping with DELOS is quite straightforward since *Actor* is mapped in the homonym class and the *Community* class is mapped into the *Group* class of DELOS. The central class of the quality domain in the 5S model is *Quality Dimension* which allows us to evaluate every major concept in a

DL [36]; this class is mapped into the *Quality Parameter* of the DELOS Reference Model. The *Service* class is mapped into the *Function* class of the DELOS Reference Model and the *Digital Object* class is mapped into the *Information Object* class by the *schema:isSimilarTo* property.

As we can see, the 5S model has no explicit representation of the architecture and policy domain of the DELOS Reference Model.

6. The User Domain

Users represent all the entities external to a DL which interact with it and are of foremost importance for a DL.

The DELOS Reference Model has a dedicated domain for users which comprehends both “humans and inanimate entities, such as software programs or physical instruments” [14]; in the 5S model, users are modeled within the “Societies” abstraction which includes “humans as well as hardware and software components, which either use or support digital library services” [13].

The RDF mapping of the user domain in the DELOS Reference Model is shown in Figure 5 and it is composed by three main classes: *Actor*, *Group* and *Role*. An *Actor* is someone or something which interacts with the DL universe, and can be either a human being or a computing device. An *Actor* is a *Resource* and inherits all its key characteristics, even if they are specialized to better fit to the notion of *Actor*. For example, the policies represent the functions that *Actors* perform or the information objects they have access to.

An *Actor* may be part of a *Group* which represents a population that has a larger cohesiveness than a single actor; a *Group*

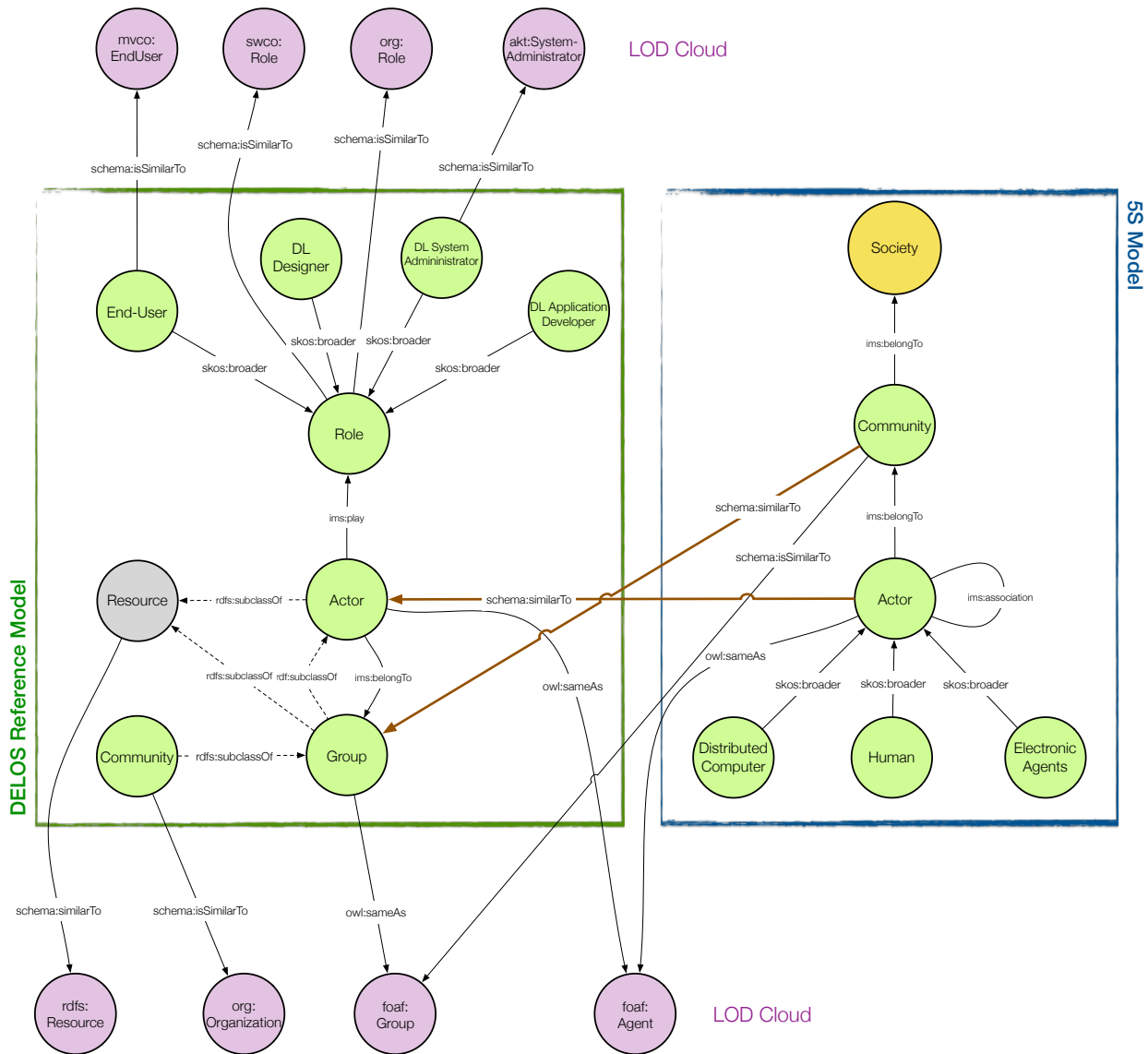


Figure 5: Semantic mapping of the user domain concepts in the 5S and the DELOS Reference models together with their relationships.

can be considered as an Actor with its own profile and identifier and it is a subclass of Actor. A subclass of Group is Community which refers to a social group of humans. As we can see in Figure 5, Actor, Group and Community are connected with some external classes of the LOD cloud by using the properties `owl:sameAs` or `schema:isSimilarTo`. For instance, Actor in DELOS is interpreted as the same (`owl:sameAs`) concept as `foaf:Agent` which is defined as “the class of agents; things that do stuff. A well known sub-class is Person, representing people.

Other kinds of agents include Organization and Group¹³ such as a person, a group, a software tool or a physical artifact. The Group class is related via an `owl:sameAs` property to the `foaf:Group` class since a group in the FOAF ontology is defined as a “concept [which] is intentionally quite broad,

covering informal and ad-hoc groups, long-lived communities, organizational groups within a workplace, etc.”¹⁴.

The `schema:isSimilarTo` property can be used to relate a Community to the class `org:Organization` which “represents a collection of people organized together into a community or other social, commercial or political structure”¹⁵.

An Actor may play different Roles at different times. The Role class is related to the `org:Role` and `swco:Role` external classes. It is possible to define a taxonomy of roles by exploiting two properties of the skos vocabulary: `skos:broader` and `skos:narrower`. Given two resources, say A and B, then A `skos:broader` B asserts that B is a broader concept than A; whereas, A `skos:narrower` B asserts that B is a narrower concept than A.

¹³http://xmlns.com/foaf/spec/#term_Agent

¹⁴http://xmlns.com/foaf/spec/#term_Group

¹⁵<http://www.w3.org/ns/org>

In Figure 5 we can see that Role can be specialized by four classes representing typical user roles in a DL, which are: End-User which defines the users exploiting DL facilities for providing, accessing, retrieving and consuming content and which is connected with the external class `mvco:EndUser`; DL Designer which exploits DLMS functions to manage, maintain and customize a DL; DL System Administrator which employs a DLMS to build a DLS which realizes a DL and relates to the external class `akt:System-Administrator`; and, DL Application Developer which defines and creates DLMS constituents by exploiting DLS and DLMS facilities. Each role can be specialized in turn by exploiting the `skos` properties; for instance, the end-user may be specialized into “content creator”, “content consumer”, “librarian”, “archivist” and so on.

In the 5S model the main class representing the user domain is Society which is composed by a set of communities (i.e. the Community class) referring to a set of individuals or Actors. Actors of the same type belong to a community and there exists a relation between them that captures “generic societal relationships” [13] between communities of actors. Actor is related to the `foaf:Agent` and Group to the `foaf:Group` external class. A community in the 5S model is as general as a group in DELOS; for this reason there is a mapping between the Community class in the 5S and the Group class in DELOS (a Community in DELOS represents a group of only humans).

User roles in the 5S model are represented as a specialization of the Actor class; the main types of actor in the 5S model are Distributed Computer, Human and Electronic Agent which, in turn, can be further specialized.

As shown in Figure 5, within the user domain, the two main bridges between DELOS and the 5S are constituted by the mapping between the corresponding Actor classes and between the Group class in DELOS and Community in the 5S.

7. The Content Domain

The content domain represents all the entities that a DL has to manage to satisfy users’ information needs. In Figure 6 we can see the RDF graph representing the content domain of the DELOS Reference Model and of the 5S model. In DELOS the central class is Information Object which is a resource and represents a unit of information within a DL that includes text documents, images, videos, sound documents and datasets. This class is related to the external class `cidoc:InformationObject` of the CIDOC-CRM ontology which “comprises identifiable immaterial items, such as a poems, jokes, data sets, images, texts, multimedia objects, procedural prescriptions, computer program code, algorithm or mathematical formulae, that have an objectively recognizable structure and are documented as single units”¹⁶. The information object is a complex one and it can capture different concepts such as the notion of edition, view and manifestation which are part of the IFLA-FRBR model [39]; these relationships are modeled by the recursive properties `hasView`, `hasManifestation` and `hasEdition`.

We can see that it is specialized via the `skos:broader` property in three main classes: (i) Primary Information Object which is an information object that stands in its own like a book or a dataset; (ii) Metadata which is an information object which gives information about a given Resource in the DL – e.g. a primary information object, an actor or a service; (iii) Annotation which enriches a Resource or a region of it with additional information – examples of annotations are links, structured comments and notes. The annotation class is related to the class annotation of the Open Annotation Data Model¹⁷ which “provides an extensible, interoperable framework for expressing annotations such that they can easily be shared between platforms”. Metadata is related to the FABIO ontology, the purpose of which is to record and publish on the Semantic Web descriptions of entities¹⁸, and its metadata class is defined as “a separate work that provides information describing one or more characteristics of a resource or entity”.

Information objects can be organized into a Collection which is a subclass of Information Object and thus the former inherits all of the modeling aspects and facilities of the latter, for instance a collection can be annotated. Collections are characterized by the property `hasIntension` relating them to the Query class; basically, a query represents a set of grouping criteria which defines the rules an information object has to respect to be part of a certain collection. We can see that the Query and the Ontology classes are related to Information Object by means of the property `expressedBy`; this property is intended for connecting a resource to an information object materializing it.

On the right hand side of Figure 6, we can see the RDF graph modeling of the 5S content domain; the main classes are Stream and Structure which define a Digital Object. Thus, a Digital Object is composed by some Streams which can be Text, Audio, Image or Video – note that Video is defined as a composition of image and audio streams – and a Structure – for instance a tree structure for representing a taxonomy or a graph for representing a hypertext. We can see that the specializations of a stream are related to several external classes in the `dcTerms` vocabulary, e.g. Audio is related to `dcTerms:Audio`. A Space is composed by a set of objects and operations on those objects and it can be associated to a Digital Object.

Collections are sets of Digital Objects stored in a Repository. The Collection class is related to the external class `fabio:Collection` and the Repository class is related to the `dctypes:Repository` external class. A repository in the 5S model stores also Metadata Catalogs which are a collection of Metadata Set grouping Metadata; the 5S defines two different classes for digital object and metadata, whereas the DELOS Reference Model considers a metadata as a specification of an information object.

The main bridges between DELOS and the 5S are represented by the `schema:isSimilarTo` properties between Digital Object in the 5S and Information Object in DELOS,

¹⁶<http://www.cidoc-crm.org/docs/>

¹⁷<http://www.openannotation.org/>

¹⁸<http://vocab.ox.ac.uk/fabio/>

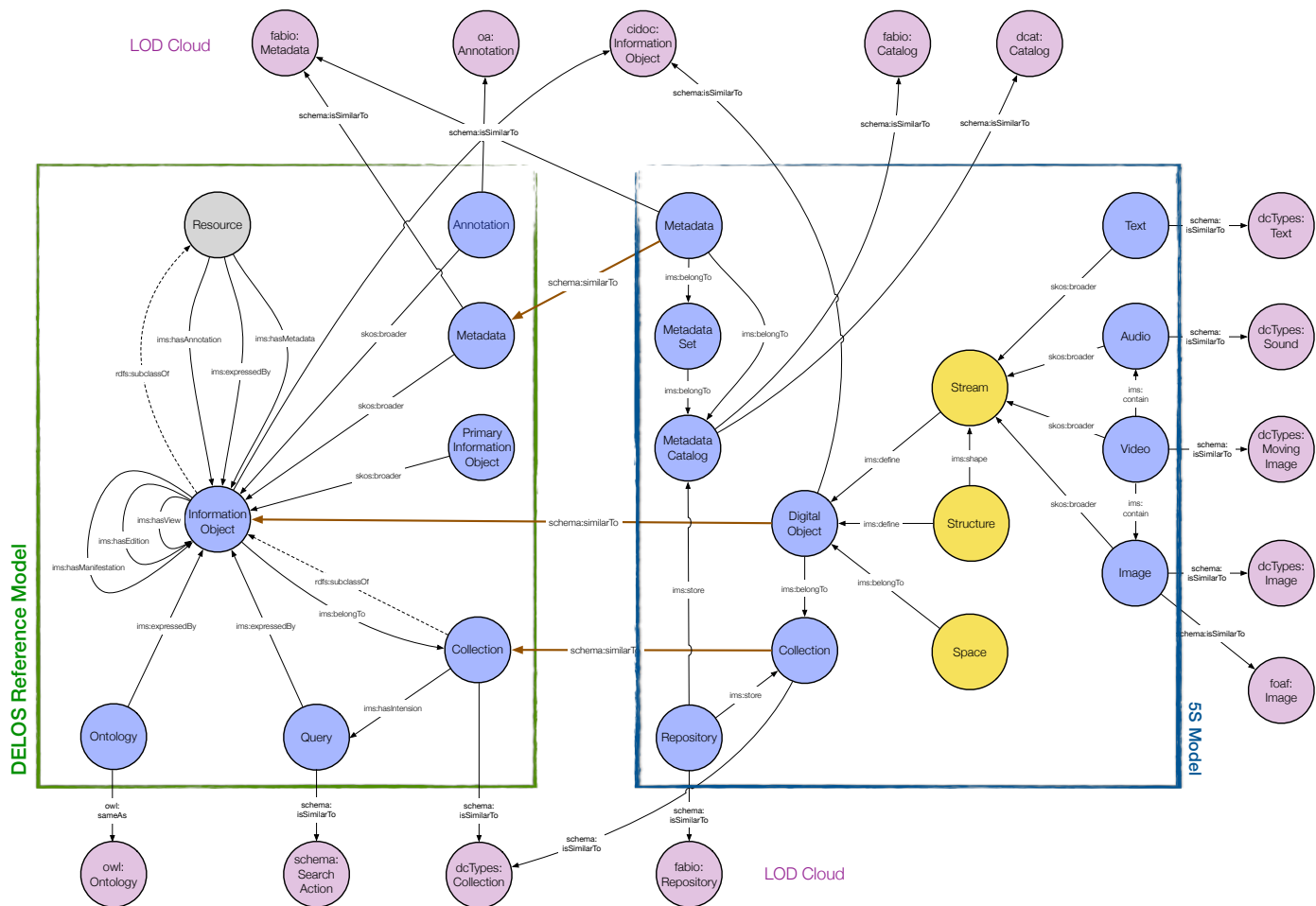


Figure 6: Semantic mapping of the content domain concepts in the 5S and the DELOS Reference models together with their relationships.

the Metadata classes and the Collection classes. These classes are also related to the same external entities in the LOD cloud – e.g. `cidoc:Information Object` for Information Object and Digital Object, `fabio:Metadata` for the Metadata classes and `dcTypes:Collection` for the Collection classes.

8. The Functionality Domain

The functionality domain represents one of the richest domains in the DL universe and it captures all the processing that can occur on resources and activities in a DL. In Figure 7 we report the RDF graph of the functionality domain for the DELOS Reference Model and the 5S model.

The DELOS Reference Model shapes this domain around the Function class defined as a particular act or operation that can be performed on a Resource (a Function actOn a Resource) by an Actor (an Actor performs a Function). It is worth noting that a Function is a Resource itself and thus, it can be described by a metadata or annotated. A function in DELOS is defined in relation to the act of an actor which may be a human, a machine or a software agent; indeed, the function of “accessing a resource” (i.e. `Access Resource`) may

be performed both by a user and a computer. A function may interactWith another function in order to build compound or complex functions.

Given the broad scope of the Function class, the DELOS Reference Model does not enumerate all possible functions of a DLS, but defines five main concepts representing a general class of activities which may be specialized into several other functions; in the RDF graph in Figure 7 they are represented by the following classes: `Access Resource` capturing all those activities related to requesting, locating and retrieving a resource; `Manage Resource` capturing those activities related to creating, updating, transforming or deleting a resource; `Manage DL` capturing all those day-to-day activities concerning a DL such as managing users, content and functions or preserving, importing and exporting collections; `Manage and Configure DLS` comprehends the functions for setting-up, configuring and monitoring the architectural components of a DLMS realizing a DL, and `Collaborate` captures all the functions that allow multiple Actors to work together.

In the 5S model the main class is `Scenario` defined as a sequence of events – an `Event isContained in a Scenario`. A scenario can be seen as a graph where nodes are states and edges are events. A set of related scenarios defines a `Service`.

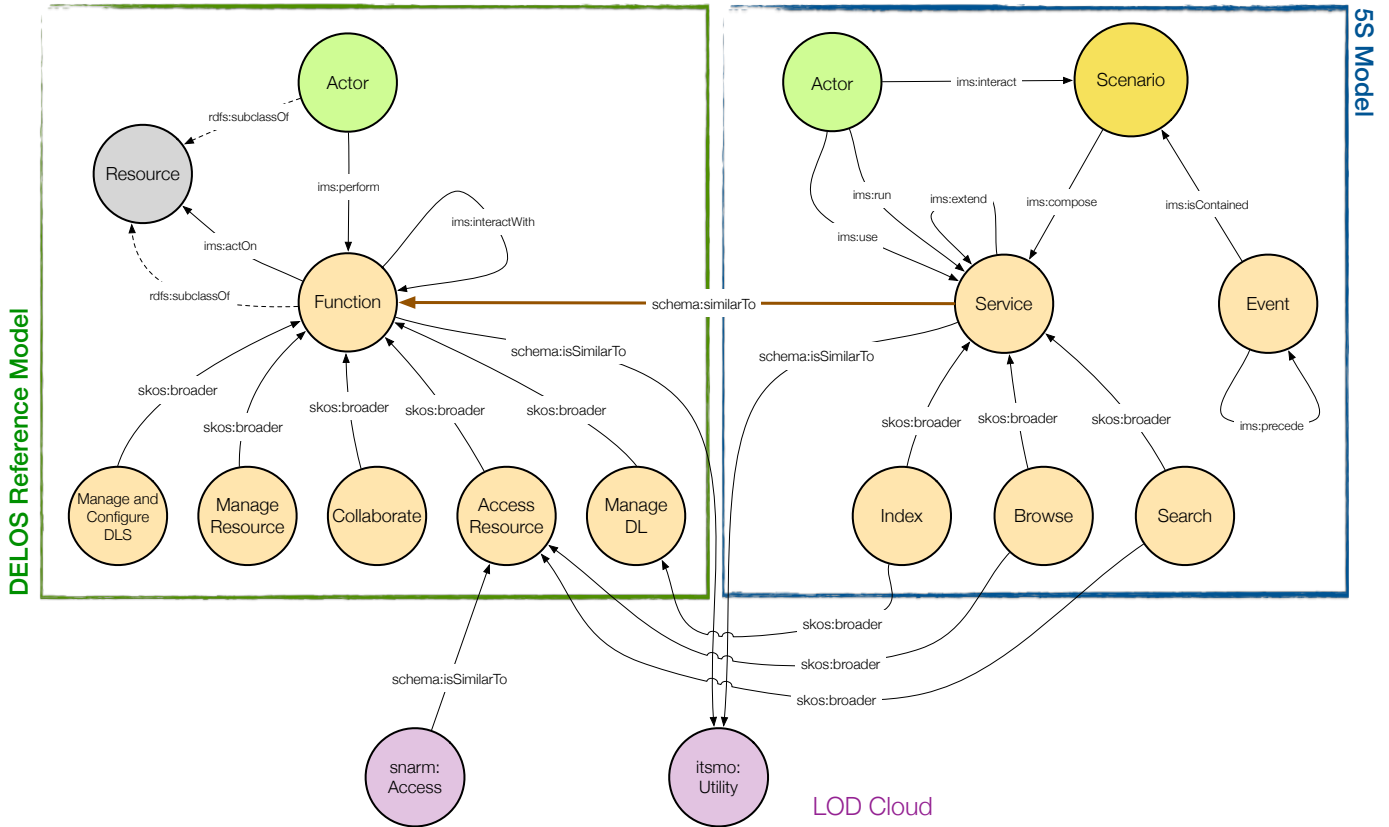


Figure 7: Semantic mapping of the functionality domain concepts in the 5S and the DELOS Reference models together with their relationships.

The precede property defines a temporal sequence of events to which an actor engages while interacting with a service.

The minimum set of services defined in the 5S are: Browse which is connected by means of a `skos:broader` property to the class Access Resource in DELOS, Index connected to Manage DL in DELOS, and Search connected to Access Resource as well.

A Service can be extended by other Services by means of the recursive property `ims:extend` and run or used by an Actor which may interact with a Scenario.

The bridge between the DELOS and the 5S RDF graphs is constituted by the `schema:isSimilarTo` property mapping the Service class into the Function one. Furthermore, these two classes are also connected via the external property `itsmo:Utility` defined as a “functionality offered by a product or service to meet a particular need. Utility can be summarized as what the service does [...]”¹⁹.

9. The Quality Domain

Quality is a fundamental aspect in DL [32, 36, 37, 40, 41], which is often related to and affected by the interoperability and integration among DL systems [15, 42, 43].

The quality domain in the DELOS Reference Model takes into account the general definition of quality provided by the

International Organization for Standardization (ISO) which defines quality as “the degree to which a set of inherent characteristics fulfills requirements” [44], where requirements are needs or expectations that are stated, generally implied or obligatory, while characteristics are distinguishing features of a product, process, or system.

A Quality Parameter is a Resource that indicates, or is linked to, performance or fulfillment of requirements by another Resource. A Quality Parameter is evaluated by a Measurement, it is measured by a Measure assigned according to the Measurement, and expresses the assessment of an Actor. With respect to the definition provided by ISO, we can note that: the “set of inherent characteristics” corresponds to the pair (Resource, Quality Parameter); the “degree of fulfillment” fits in with the pair (Measurement, Measure); finally, the “requirements” are taken into consideration by the assessment expressed by an Actor.

Quality can be characterized by various Quality Parameters, each capturing how the Resource performs with respect to some attribute; it is regulated by policies governing every aspect of its lifetime; it is expressed by an information object; and, it can be described by or commented on by an information object, especially by metadata and annotations.

The resources defined in the quality domain can be represented as *subclasses* of Resource. The main classes are: Quality Parameter, Measure and Measurement and they are depicted in Figure 8.

¹⁹<http://ontology.it/itsmo/v1/itsmo.html#term.Utility>

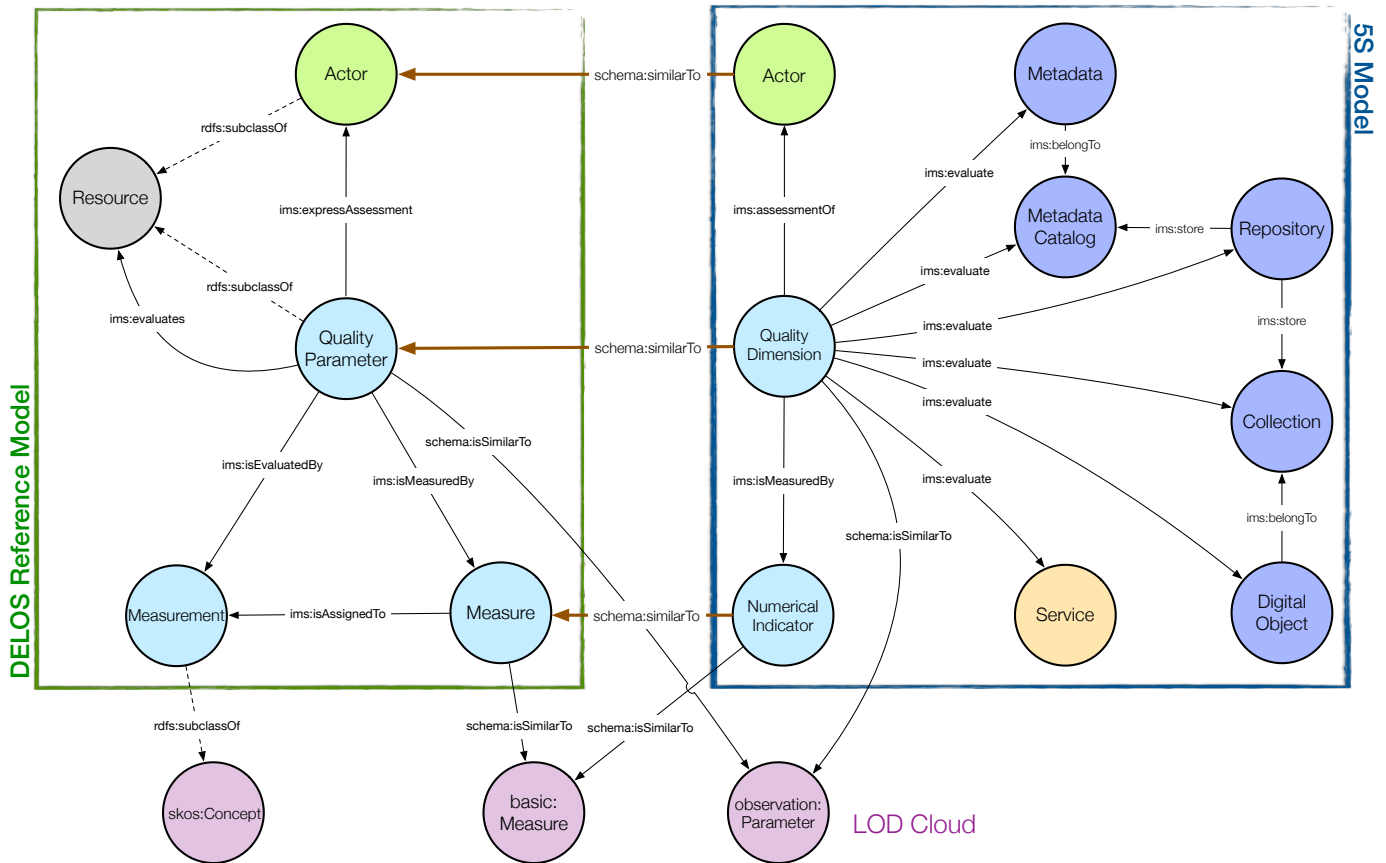


Figure 8: Semantic mapping of the quality domain concepts in the 5S and the DELOS Reference models together with their relationships.

The Measure class is similar to the `basic:Measure` class in the OWL representation of ISO 19103²⁰ which is defined as a scaled number with a unity of measure. Also the Quality Parameter class can be related to an external class by means of the `schema:isSimilarTo` property; as we can see in Figure 8 it can be related to the `observation:Parameter` class in the OWL representation of ISO 19156 (Observation model)²¹.

We define Measurement as a subclass of the `skos:Concept` which is defined as an idea or notion, a unit of thought. Usually, `skos:Concept` is used to define the type of relationships in a semantic environment or to create a taxonomy [45, 46].

Quality Parameters serve the purpose of expressing the different facets of the quality domain. In this model, each Quality Parameter is itself a Resource and inherits all its characteristics, as, for example, the property of having a unique identifier. Quality Parameters provide information about how, and how well, a Resource performs with respect to some viewpoint and resemble the notion of quality dimension in [47]. They express the assessment of an Actor about the Resource under examination. They can be evaluated according to different Measurements, which provide alternative procedures for assessing different aspects of a Quality Parameter and assigning it a

value, i.e. a Measure. Being a Resource, a Quality Parameter can be organized in arbitrarily complex and structured forms because of the composition and linking facilities, e.g. a Quality Parameter can be the compound of smaller Quality Parameters each capturing a specific aspect of the whole or it can be itself characterized and affected by various Quality Parameters. For example, Availability is affected by Robustness and Fault Management: in fact, when a function is both robust and able to recover from error conditions, it is probable that its availability is also increased.

A Quality Parameter can be regulated or affected by policies. For example, the Economic Convenience of accessing a DL may be affected by its charging policy, since the latter is responsible for the definition of the charging strategies adopted by the DL. Finally, a Quality Parameter can be enriched with metadata and annotations. In particular, the former can provide useful information about the provenance of a Quality Parameter, while the latter can offer the possibility to add comments about a Quality Parameter, interpreting the obtained values, and proposing actions to improve it. In order to clarify the relationship between Quality Parameter, Measurement and Measure, we can take an example from the information retrieval field. One of the main Quality Parameters of an information retrieval system is its effectiveness, meant as its capability to answer user information needs with relevant items. This Quality Parameter can be evaluated ac-

²⁰<http://def.seegrid.csiro.au/isotc211/iso19103/2005/basic#Measure>

²¹<http://def.seegrid.csiro.au/isotc211/iso19156/2011/observation>

ording to many different Measurements, such as precision and recall [48]: precision evaluates effectiveness in the sense of the ability of the system to reject useless items, while recall evaluates effectiveness in the sense of the ability of the system to retrieve useful items. The actual values for precision and recall are Measures and are usually computed using standard tools, such as `trec_eval`²², which are Actors, but in this case not human.

The quality domain in the 5S model, proposed in [36], is defined around the Quality Dimension class which evaluates a bunch of key concepts defining a minimal DL: Digital Object, Metadata, Metadata Catalog, Collection, Repository, and Service.

The Quality Dimension class can be specified into a taxonomy of dimensions related to a specific object of evaluation; for instance, the minimal set of dimensions evaluating Digital Object comprehends: accessibility, pertinence, preservability, relevance, similarity, significance and timeliness. Each quality dimension is measuredBy a Numerical Indicator and expressed by an Actor.

The mapping between DELOS and the 5S is realized by the schema: `isSimilarTo` properties between Quality Dimension and Quality Parameter and Numerical Indicator and Measure.

10. The Policy Domain

The policy domain represents a set of conditions, rules or regulations governing a DL and it is a domain “very broad and dynamic by nature” [14]. The domain is modeled by DELOS model, whereas it is not explicitly handled by the 5S. Thus, for this domain it is not possible to provide a straightforward mapping between the two models, but we can use the common ontology we defined to enrich the 5S with some concepts of DELOS and at the same time use the policy domain as a bridge to further connect the two models.

In the DELOS reference model, a Policy can be classified into Characteristic Policy which details the characteristics of a policy (e.g. prescriptive or descriptive policy) and Scope Policy which regards a specific aspect of a DL such as system, content, user and functionality.

As we can see in Figure 9 the class Policy represents a concept which is actually realized by an Information Object and directly connected to a Digital Object in the 5S model and thus can be expressed by a policy in DELOS – i.e., by a Content Policy. In Figure 6 we can see that also Metadata and Collection in the 5S are directly connected to corresponding classes in DELOS and thus can be expressed by policies.

In the 5S the idea of policy is present as connected to the concept of Society given that it “exists [also] to describe the context of [a DL] use” [13], for example, “policies may dictate that only certain communities have the right to use specific portions of a collection” [13]. Furthermore, in the 5S policies are

considered also in relation to Services and Streams since policies “can be enforced by specific services – e.g. authentication, authorization, and specific practices (scenarios) or protocols, which can involve other communication services and serialized streams” [13].

Thus, within this domain the principal link between DELOS and the 5S passes through the User Policy which regulates an Actor or a Group; an Actor in DELOS is directly connected to the Actor class in the 5S as well as a Group is connected to a Community as shown in Figure 5. In the same vein, a Functionality Policy which regulates a Function in DELOS, can regulate a Service in the 5S given that these two classes are directly connected as shown in Figure 7.

The Policy class is connected with two external classes from the LOD cloud: `dcTerms:Policy` and `fabio:Policy`. These classes can act as a bridge towards other models or DL. A DL built following the 5S model may employ some policy modeled accordingly to some class in the LOD cloud and possible connections with the classes connected to the DELOS Policy class may be used a bridge between the 5S and DELOS.

11. The Architectural Component Domain

The architectural component domain regards concepts and relationships characterizing a DLS and a DLMS in the DELOS reference model. In general, this domain aims at modeling the components of the software systems employed in a DL universe and their interactions. In the DELOS reference model an architectural component is connected to a functionality meaning that a specific function, e.g. searching, is realized by a specific architectural component, i.e. search engine, in a given DLS. On the other hand, this aspect is not explicitly modeled in the 5S model even though it is present and as important as in DELOS; in the 5S the architectural component domain is enclosed in the function domain. This means that the Service class in the 5S comprises both the a functionality and the architectural components realizing it.

On the left-hand side of Figure 10 we can see the main classes composing this domain in the DELOS reference model. The main class is Architectural Component which yield a Function. An architectural component is specialized by two main classes Software Architectural Component and System Architectural Component describing a software and a system architecture respectively. A software architectural component can be specialized into a Software Component, an Interface and a Framework Specification. The DELOS reference model dedicates particular attention to software components which encapsulate the implementation of a portion of a software system and their usages are regulatedBy a License (i.e. a specification of Policy). Note that a software component is realized by a Running Component which is a running instance of the software component.

In Figure 7 we have seen that the Function class in DELOS is directly connected to the Service class in the 5S. In DELOS an architectural component yield a Function, thus an architectural component may yield a Service in the 5S. Furthermore, we have seen that a Policy of DELOS can be

²²http://trec.nist.gov/trec_eval/

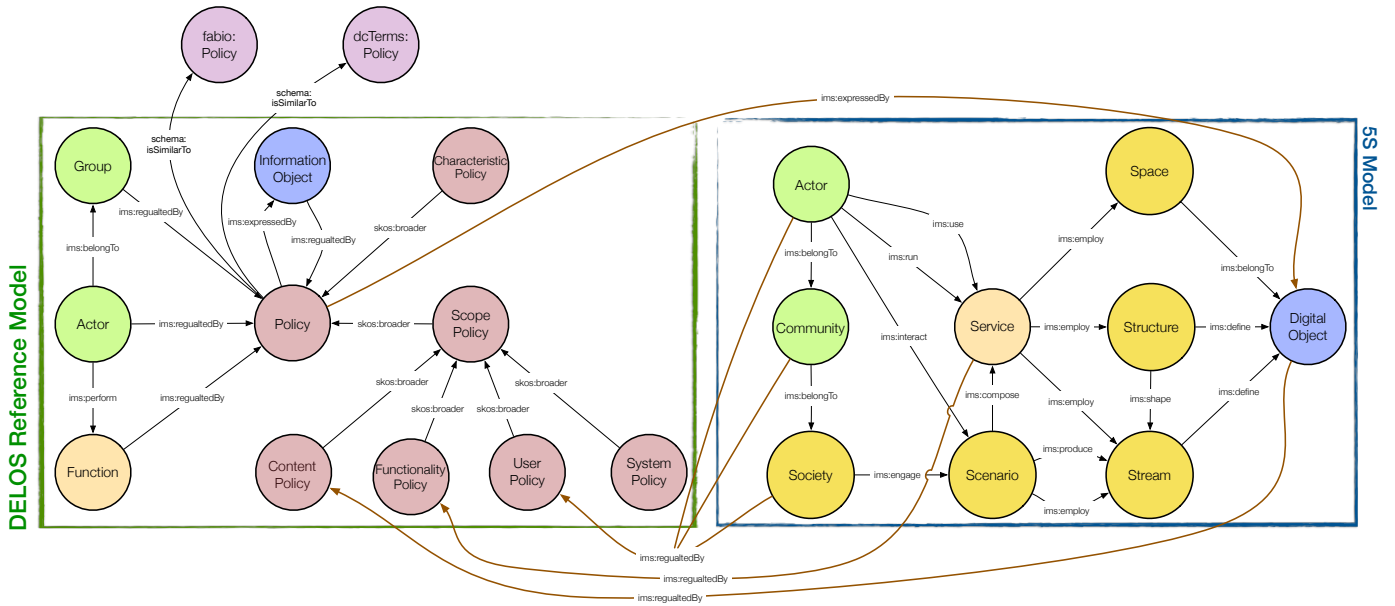


Figure 9: Semantic mapping of the policy domain concepts in the 5S and the DELOS Reference models together with their relationships.

expressed by a Digital Object in the 5S, thus a 5S Service can be regulatedBy a DELOS policy. We see that being able to connect 5S services to DELOS functions creates further bridges between the two models.

By means of the proposed ontology a DELOS architectural component could be used by a 5S service or a 5S service could be specified by several DELOS components. On the other hand, a 5S service may also be an architectural component itself given that this class in the 5S has a wider meaning than the corresponding one (i.e. Function) in the DELOS Reference Model.

As we can see in Figure 10, several classes of this domain are connected to external entities in the LOD cloud and they can serve as further bridges between the models.

12. Use Case

Let us consider the case of two different DLS managing cultural heritage digital resources and providing annotation services to their users.

The first system, called DLS1, is the DelosDLMS [49, 50], which implements the DELOS Reference Model, offers annotation functionalities through the *Flexible Annotation Semantic Tool (FAST)* service and exploits them to improve search and retrieval of annotated resources [51–53]. The second system, DLS2, is an implementation of the 5S model.

According to the FAST annotation model [11], an annotation is a compound multimedia object which is constituted by different signs of annotation. Each sign materializes part of the annotation itself; for example, we can have textual signs, which contain the textual content of the annotation, image signs, if the annotation is made up of images, and so on. In turn, each sign is characterized by one or more meanings of annotation, which specify the semantics of the sign; for example, we can have a sign whose meaning corresponds to the title field in the

Dublin Core (DC) metadata schema, in the case of a metadata annotation, or we can have a sign carrying a question of the authors about a document whose meaning may be “question” or similar. Moreover, an annotation has a scope which defines its visibility (public, shared, or private), and can be shared with different groups of users. Public annotations can be read by everyone and modified only by their owner; shared annotations can be modified by their owner and accessed by the specified list of groups with the given access permissions, e.g. read only or read/write; private annotations can be read and modified only by their owner.

Figure 11 shows a real-world example based on annotations of two illuminated manuscripts (i.e. herbals) and summarizes the discussion so far [54]. The annotation, with identifier a1, is authored by the user u1. It annotates an illustration from the *Carrarese Herbal*, whose identifier is dls1.org/manuscript/135/page/12, managed by DLS1. The annotation relates to another illustration from the *Roccabonella Herbal*, whose identifier is dls2.org/herbal/roccabonella/fig4, managed by DLS2; in addition, it relates to the DBpedia page of the Carraresi family, <http://dbpedia.org/resource/Carraresi>, which endorsed the production of the *Carrarese Herbal*. In particular, a1 annotates a region of the *Carrarese Herbal* representing a cucumber by using a textual sign whose content is “This illumination presents an extraordinary search for realism” and whose meaning is comment in the RDFS namespace [55]. a1 relates the *Carrarese Herbal* managed by DLS1 to the *Roccabonella Herbal* managed by DLS2. In particular it points to a region of an illustration representing a cucumber as well, with a textual sign whose content is “The Roccabonella Herbal illumination is clearly copied from the Carrarese Herbal one, as it shows the same disposition of the elements of the plant in the page, the same search for realism and the same attention to the light effects on the surface of the leaves, the fruits and

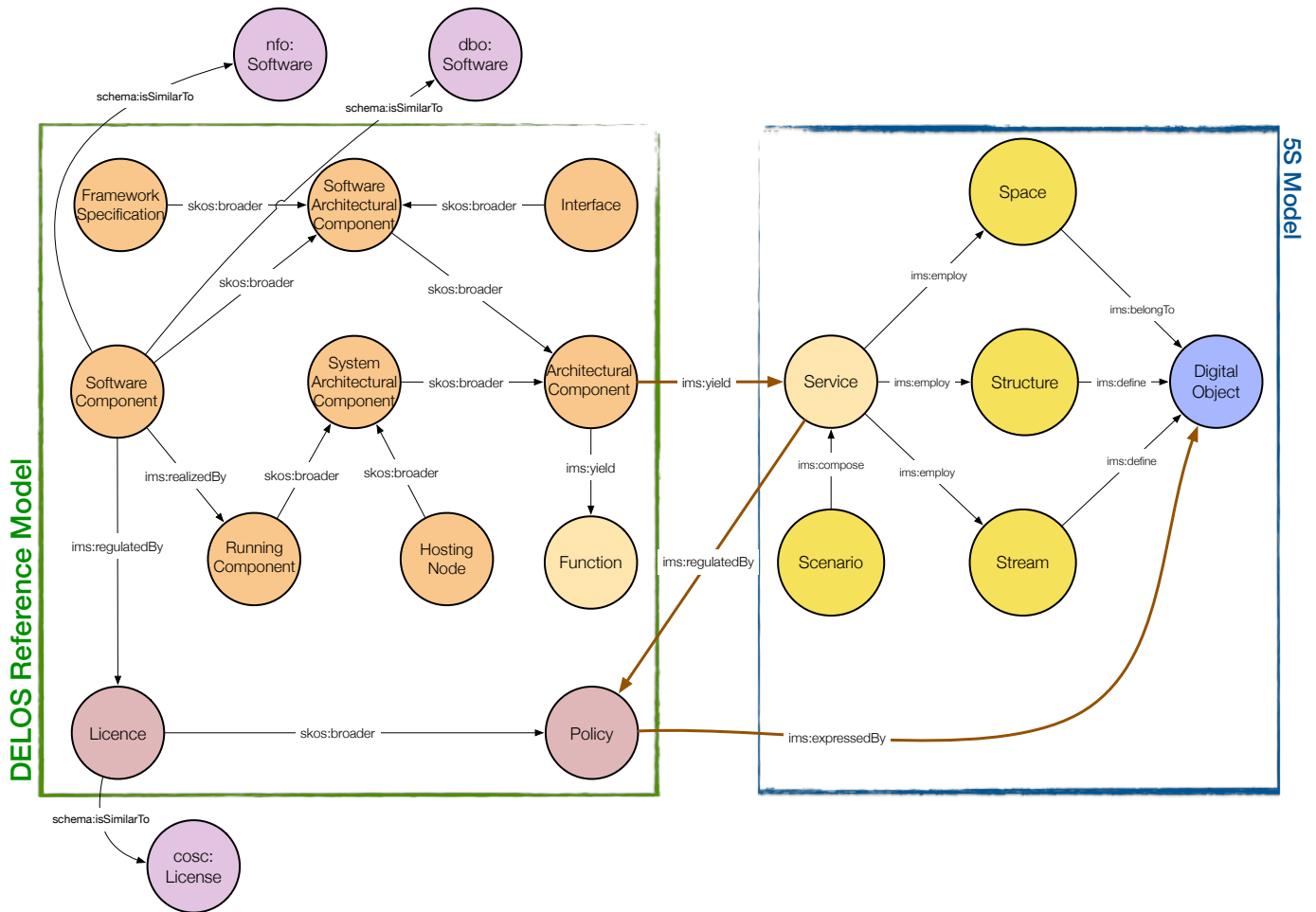


Figure 10: Semantic mapping of the architectural component domain concepts in the 5S and the DELOS Reference models together with their relationships.

the flowers” and whose meaning is `isVersionOf` in the DCMI Metadata Terms namespace²³.

Therefore, this annotation may represent the outcomes of the actual work of a historian of art, who conducted his/her own research on these two herbals, to determine that one was inspired or copied from the other one.

Moreover, `a1` relates the *Carrarese Herbal* to the DBpedia page of the Carraresi family, which endorsed the herbal, with an image sign with a picture of a building of the Carraresi family, whose meaning is `seeAlso` in the RDFS namespace.

Now, let us suppose that to another historian of art, say user `u2`, who is working on the image of the *Roccabonella Herbal* managed by DLS2, the URI of annotation `a1` is recommended as relevant to her/his work. The need of `u2` is to read `a1`, thus by resolving the URI of `a1`, DLS2 understands that `a1` is managed by DLS1 which in turn recognizes it to be an Annotation. At this point, the common ontology mapping the 5S model into the DELOS Reference Model (whose relevant portion to this example is shown in Figure 12) comes into play, because it allows DLS2 to understand that an Annotation in DELOS is an Information Object corresponding to a Digital Object

in the 5S. Furthermore, the common ontology allows DLS2 to see that `Read Annotation` is a particular specialization of `Access Resource` which is the `Function` of DLS1 which acts on annotations; the `Browse service` of DLS2 is a specialization of `Access Resource` and a `Function` in DLS1 is similar to a `Service` in DLS2. Once DLS2 has all this information, it knows which kind of object `a1` is and it knows how to interact with it, thus allowing `u2` to read the annotation.

Since `Annotation` relates to `oa:Annotation` in the Open Annotation Model, other systems based on this model, such as Web browsers, can interoperate with DLS1 and DLS2 because, exploiting the “DL Foundational Models Ontology”, they can represent their notion of annotation as an `oa:Annotation` for these external systems.

The access of Actor `u2` to annotation `a1` in DLS1 is regulated by an `Access Policy` expressed by `Information Object` `p1` which is related to `Digital Object` `p1` in DLS2 by means of the `similar` property. As we have seen in Section 10, it is possible to establish a bridge between the DELOS Reference Model and the 5S Model also in the policy domain, even though this is not explicitly modeled by the 5S model. Indeed, the 5S model implicitly comprises the concept of policy, which is materialized by a digital object in DLS2 and can be directly

²³<http://dublincore.org/documents/dcmi-terms/>

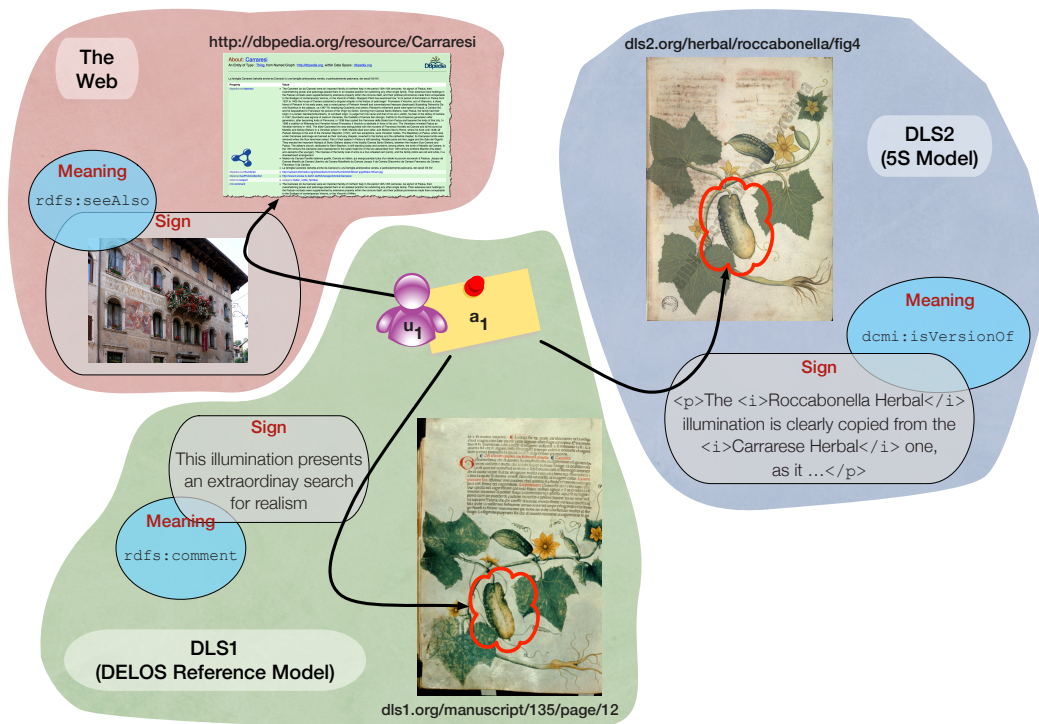


Figure 11: Example of annotation spanning different DLS and the Web.

connected to the similar information object in DLS1. The connection between the two DLS can be enabled also by exploiting external classes in the LOD cloud; in fact, the Digital Object p_1 in DLS2 can be connected to an external class such as `dcTerms:Policy` by means of a `schema:similarTo` property. Given that also the Information Object p_1 in DLS1 can be connected to the same external class, this bridge enables a further interoperability possibility between the two DLS.

Finally, in Section 11 we have seen that a Service in the 5S Model can represent a Function or an Architectural Component in the DELOS Reference Model. In Figure 12, we see that an Architectural Component in DLS1 can yield a Service in DLS2 as well as that a specific Service in DLS2 is similar to a Function in DLS1.

13. Conclusions

In this paper we address, for the first time, the need for interoperability among DLS at a high level of abstraction and we show how this is achieved by a semantically-enabled representation of foundational DL models. The ultimate goal is to promote and facilitate a better convergence and integration in the context of *Libraries, Archives and Museums (LAM)* by lowering the barriers between them.

We show: (i) how the two state-of-the-art foundational models, namely the DELOS Reference Model and the 5S model, can be represented by using Semantic Web technologies; (ii) how they can be mapped one into another; and, (iii) how their concepts can be linked to the others in the LOD cloud in order to attain maximum interoperability.

We propose a common ontology which encompasses all the concepts considered by the two foundational models and creates explicit connections between their constituent domains. In particular, the user, functionality and content domains allow us to enable a high-level interoperability between the actors and the information/digital objects of DL as well as their functions/services. The quality domain, which is fundamental and often related to and affected by the interoperability and integration among DL systems, is extensively treated and several connections between the models are pointed out. Finally, the Policy and Architecture domains regulate the actual implementation of a DLS.

The DELOS Reference Model and the 5S Model are defined starting from two different viewpoints. Indeed, in the DELOS Reference Model the approach is top-down since it defines the entities and relationships involved in a DL trying to be as extensive as possible; whereas the 5S model is largely bottom-up starting with key definitions and elucidation of digital library concepts from a minimalist approach. For this reason, some of the concepts modeled by the DELOS Reference Model are not explicitly modeled by the 5S model; for instance, this is the case of the policy domain and the architectural component domain which are modeled by the DELOS Reference Model but not by the 5S one. The common ontology we define is particularly effective also in this case since it enriches the 5S model with the concepts defined by the DELOS Reference Model, creating further bridges between them and their implementations.

In particular, we highlight the important role of *service* in the 5S Model which is a broad concept comprising both the concepts of *function* and *architectural component* in the DELOS Reference Model. The presented ontology explicitly points out

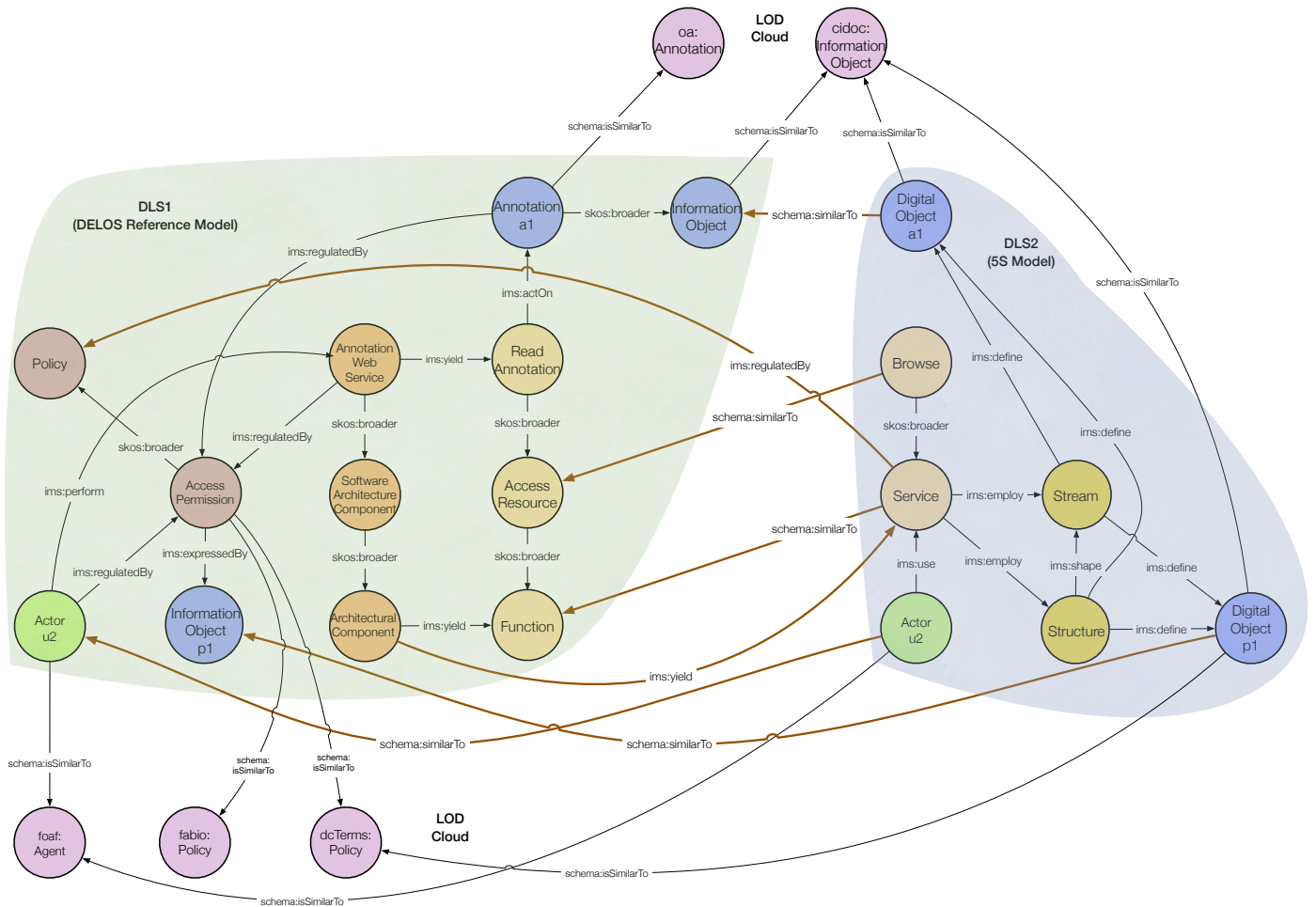


Figure 12: Interoperability between two different DLS modeled according to DELOS Reference Model and the 5S model.

the connection between these concepts enabling connections between the two models which were not easy to recognize and establish otherwise. Furthermore, we investigated how policies are modeled and used by the two models; we pointed out that in the 5S model, the idea of *policy* is encompassed by the concept of *Society* even though it is not explicitly treated. Starting from this consideration, the common ontology enables the possibility of using policies defined in the DELOS Reference Model to regulate services in the 5S.

Finally, we presented a concrete use case based on two DLS modeled according to the DELOS Reference Model and the 5S Model which manage and provide access to digital annotations. Even though this use case focuses on a quite specific aspect covered by two DLS, it allows us to see how many classes are involved in the process and how many possible interoperability issues may arise in practice. In this case the ontology proves itself to be powerful and wide enough to enable an actual interoperability between the two considered DLS; indeed, it allows a user of DLS2 to read (i.e. use a function) an annotation of DLS1 according to its internal policies regulating access permissions.

The ISO/IEC 2382-2015 Information Technology Vocabulary, defines interoperability as “the capability to communi-

cate, execute programs, or transfer data among various functional units in a manner that requires minimal knowledge of the unique characteristics of those units” [56]. As you can note, the ISO definition contains all the main features needed to characterize interoperability from a general point of view but, as a consequence, it lacks the contextualization necessary to apply it to a specific domain, as the DL one can be. This paper moves a substantial step forward in making interoperability among DL closer to the ISO definition, since it provides the high level concepts needed to communicate, execute programs or transfer data, by representing the knowledge required in a minimal and consistent manner.

However, much work is still ahead of us, since the proposed ontology needs to be operationalized into actual DLS and, probably, it will need to be extended both to accomplish specific details that arise when you make actual systems interoperate and to address peculiar needs of specialised domains, which may depart from the common general view.

References

- [1] E. A. Fox, D. Hix, L. T. Nowell, D. J. Brueni, W. C. Wake, L. S. Heath, D. Rao, Users, User Interfaces, and Objects: Envision, a Digital Library,

- Journal of the American Society for Information Science (JASIS) 44 (8) (1993) 480–491.
- [2] E. A. Fox, R. M. Aksyn, R. K. Furuta, J. J. Leggett, Digital Libraries, Communications of the ACM (CACM) 38 (4) (1995) 22–28.
 - [3] G. Marchionini, H. Maurer, The Roles of Digital Libraries in Teaching and Learning, Communications of the ACM (CACM) 38 (4) (1995) 67–75.
 - [4] M. Lesk, Practical Digital Libraries. Books, Bytes & Bucks, Morgan Kaufmann Publishers, San Francisco, California, USA, 1997.
 - [5] C. L. Borgman, What are digital libraries? Competing visions, Information Processing & Management 35 (3) (1999) 227–243.
 - [6] C. L. Borgman, From Gutenberg to the Global Information Infrastructure: Access to Information in the Networked World (Digital Libraries and Electronic Publishing), The MIT Press, Cambridge (MA), USA, 2003.
 - [7] I. H. Witten, D. Bainbridge, How to Build a Digital Library, Morgan Kaufmann Publishers, San Francisco (CA), USA, 2003.
 - [8] E. A. Fox, M. A. Gonçalves, R. Shen, Theoretical Foundations for Digital Libraries: The 5S (Societies, Scenarios, Spaces, Structures, Streams) Approach, Morgan & Claypool Publishers, USA, 2012.
 - [9] D. M. Zorich, G. Waibel, R. Erway, Zorich, Beyond the Silos of the LAMs. Collaboration Among Libraries, Archives and Museums, OCLC Research, Dublin, USA. <http://www.oclc.org/content/dam/research/publications/library/2008/2008-05.pdf> (September 2008).
 - [10] M. Agosti, Digital Libraries, in: M. Melucci, R. A. Baeza-Yates (Eds.), Advanced Topics in Information Retrieval, The Information Retrieval Series, Vol. 33, Springer-Verlag, Heidelberg, Germany, 2011, pp. 1–26.
 - [11] M. Agosti, N. Ferro, A Formal Model of Annotations of Digital Content, ACM Transactions on Information Systems (TOIS) 26 (1) (2008) 3:1–3:57.
 - [12] Y. Ioannidis, D. Maier, S. Abiteboul, P. Buneman, S. Davidson, E. A. Fox, A. Halevy, C. Knoblock, F. Rabbitt, H.-J. Schek, G. Weikum, Digital Library Information-Technology Infrastructures, International Journal on Digital Libraries 5 (4) (2005) 266–274.
 - [13] M. A. Gonçalves, E. A. Fox, L. T. Watson, N. A. Kipp, Streams, Structures, Spaces, Scenarios, Societies (5S): A Formal Model for Digital Libraries, ACM Transactions on Information Systems (TOIS) 22 (2) (2004) 270–312.
 - [14] L. Candela, D. Castelli, N. Ferro, Y. Ioannidis, G. Koutrika, C. Meghini, P. Pagano, S. Ross, D. Soergel, M. Agosti, M. Dobrev, V. Katifori, H. Schuld, The DELOS Digital Library Reference Model. Foundations for Digital Libraries, ISTI-CNR at Gruppo ALI, Pisa, Italy, http://delosw.isti.cnr.it/files/pdf/ReferenceModel/DELOS_DLReferenceModel.0.98.pdf, 2007.
 - [15] N. Ferro, Quality and Interoperability: The Quest for the Optimal Balance, in: I. Iglezakis, T.-E. Synodinou, S. Kapidakis (Eds.), E-Publishing and Digital Libraries: Legal and Organizational Issues, IGI Global, USA, 2010, pp. 48–68.
 - [16] T. Heath, C. Bizer, Linked Data: Evolving the Web into a Global Data Space, Morgan & Claypool Publishers, USA, 2011.
 - [17] E. Hyvönen, Publishing and Using Cultural Heritage Linked Data on the Semantic Web, Morgan & Claypool Publishers, USA, 2012.
 - [18] T. Baker, E. Bermès, K. Coyle, G. Dunsire, A. Isaac, P. Murray, M. Panzer, J. Schneider, R. Singer, E. Summers, W. Waites, J. Young, M. Zeng, Library Linked Data Incubator Group Final Report – W3C Incubator Group Report 25 October 2011, <http://www.w3.org/2005/Incubator/11d/XGR-11d/> (October 2011).
 - [19] ISO 15836:2009, Information and Documentation - The Dublin Core Metadata Element Set, Recommendation ISO 15836:2009 (2009).
 - [20] Society of American Archivists, Encoded Archival Description: Tag Library, ver. 2002, Society of American Archivists, <http://www.loc.gov/ead/tglib/>, 2003.
 - [21] E. Coburn, R. Light, G. McKenna, R. Stein, A. Vitzthum, LIDO - Lightweight Information Describing Objects – Version 1.0, <http://www.lido-schema.org/schema/v1.0/lido-v1.0-specification.pdf> (November 2010).
 - [22] C. Lagoze, H. Van De Sompel, P. Johnston, M. Nelson, R. Sander-son, S. Warner, ORE Specification – Abstract Data Model – Version 1.00, <http://www.openarchives.org/ore/1.0/datamodel> (October 2008).
 - [23] ISO 21127, Information and documentation – A reference ontology for the interchange of cultural heritage information, Recommendation ISO 21127:2014 (2014).
 - [24] M. Doerr, S. Gradmann, S. Henicke, A. Isaac, C. Meghini, H. Van de Sompel, The Europeana Data Model (EDM), in: IFLA 2011: World Library and Information Congress: 76th IFLA General Conference and Assembly, Gothenburg, Sweden, 2010.
 - [25] Europeana, Europeana Data Model Primer, <http://pro.europeana.eu/documents/900548/770bdb58-c60e-4beb-a687-874639312ba5> (October 2011).
 - [26] Europeana, Definition of the Europeana Data Model Elements – Version 5.2.3, 24/02/2012, <http://pro.europeana.eu/documents/900548/bb6b51df-ad11-4a78-8d8a-44cc41810f22> (February 2012).
 - [27] C. Lagoze, H. Van De Sompel, M. Nelson, S. Warner, The Open Archives Initiative Protocol for Metadata Harvesting – Version 2.0, <http://www.openarchives.org/OAI/openarchivesprotocol.html> (December 2008).
 - [28] W3C, SOAP Version 1.2 Part 1: Messaging Framework (Second Edition) – W3C Recommendation 27 April 2007, <http://www.w3.org/TR/soap12-part1/> (April 2007).
 - [29] W3C, Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language – W3C Recommendation 26 June 2007, <http://www.w3.org/TR/wsdl20/> (June 2007).
 - [30] W3C, Semantic Annotations for WSDL and XML Schema – W3C Recommendation 28 August 2007, <http://www.w3.org/TR/sawsdl/> (August 2007).
 - [31] N. Ferro, G. Silvello, Towards a Semantic Web Enabled Representation of DL Foundational Models: The Quality Domain Example, in: D. Calvanese, C. Tasso (Eds.), Proc. 11th Italian Research Conference on Digital Libraries (IRCDL 2015), 2015.
 - [32] M. A. Gonçalves, E. A. Fox, L. T. Watson, Towards a Digital Library Theory: a Formal Digital Library Ontology, International Journal on Digital Libraries 8 (2) (2008) 91–114.
 - [33] J. D. Novak, Concept Maps and Vee Diagrams: Two Metacognitive Tools to Facilitate Meaningful Learning, Instructional Science 19 (1) (1990) 29–52.
 - [34] J. D. Novak, A. J. Cañas, The Theory Underlying Concept Maps and How to Construct and Use Them, Technical Report IHMC CmapTools 2006-01 Rev 2008-01, Florida Institute for Human and Machine Cognition (FI), USA (2008).
 - [35] M. A. Gonçalves, E. A. Fox, 5SL - A Language for Declarative Specification and Generation of Digital Libraries, in: W. Hersh, G. Marchionini (Eds.), Proc. 2nd ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2002), ACM Press, New York, USA, 2002, pp. 263–272.
 - [36] M. A. Gonçalves, B. Lagoeiro, E. A. Fox, L. T. Watson, What is a Good Digital Library? A Quality Model for Digital Libraries, Information Processing & Management 43 (5) (2007) 1416–1437.
 - [37] B. L. Moreira, M. A. Gonçalves, A. H. F. Laender, E. A. Fox, Automatic Evaluation of Digital Libraries with 5SQual, Journal of Informetrics 3 (2) (2009) 102–123.
 - [38] W3C, Architecture of the World Wide Web, Volume One – W3C Recommendation 15 December 2004, <http://www.w3.org/TR/webarch/> (December 2004).
 - [39] IFLA Study Group on the Functional Requirements for Bibliographic Records (Ed.), Functional Requirements for Bibliographic Records, no. 19 in IFLA Series on Bibliographic Control, Munich: K.G. Saur Verlag, 1998.
 - [40] M. Agosti, N. Ferro, E. A. Fox, M. A. Gonçalves, B. Lagoeiro, Towards a Reference Quality Model for Digital Libraries, in: D. Castelli, E. A. Fox (Eds.), Pre-proceedings of the First International Workshop on Foundations of Digital Libraries, 2007, pp. 37–42.
 - [41] N. Fuhr, G. Tsakonias, T. Aalberg, M. Agosti, P. Hansen, S. Kapidakis, C.-P. Klas, L. Kovács, M. Landoni, A. Micsik, C. Papatheodorou, C. Peters, I. Sølberg, Evaluation of Digital Libraries, International Journal on Digital Libraries 8 (1) (2007) 21–38.
 - [42] R. Shen, M. A. Gonçalves, E. A. Fox, Key Issues Regarding Digital Libraries. Evaluation and Integration, Morgan & Claypool Publishers, USA, 2013.
 - [43] G. Vullo, G. Clavel, N. Ferro, S. Higgins, R. van Horik, W. Horstmann, S. Kapidakis, S. Ross, Quality Interoperability within Digital Libraries: the DL.org Perspective, in: D. Castelli, Y. Ioanni-

- dis, S. Ross (Eds.), Pre-proceedings of the Second DL.org Workshop – Making Digital Libraries Interoperable: Challenges and Approaches, DL.org, <http://www.dl.org.eu/uploads/Booklets/2nd%20Workshop%20Proceedings/Pre-proceedings-1.pdf>, 2010, pp. 12–24.
- [44] ISO 9000, Quality Management Systems – Fundamentals and Vocabulary, Recommendation ISO 9000:2005 (2005).
- [45] W3C, SKOS Simple Knowledge Organization System Reference – W3C Recommendation 18 August 2009, <http://www.w3.org/TR/skos-reference> (August 2009).
- [46] W3C, SKOS Simple Knowledge Organization System Primer – W3C Working Group Note 18 August 2009, <http://www.w3.org/TR/skos-primer> (August 2009).
- [47] C. Batini, M. Scannapieco, Data Quality. Concepts, Methodologies and Techniques, Springer-Verlag, Heidelberg, Germany, 2006.
- [48] G. Salton, M. J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, New York, USA, 1983.
- [49] M. Agosti, S. Berretti, G. Brettlecker, A. del Bimbo, N. Ferro, N. Fuhr, D. Keim, C.-P. Klas, T. Lidy, D. Milano, M. Norrie, P. Ranaldi, A. Rauber, H.-J. Schek, T. Schreck, H. Schuldt, B. Signer, M. Springmann, Delos- DLMS – the Integrated DELOS Digital Library Management System, in: C. Thanos, F. Borri, L. Candela (Eds.), Digital Libraries: Research and Development. First International DELOS Conference. Revised Selected Papers, Lecture Notes in Computer Science (LNCS) 4877, Springer, Heidelberg, Germany, 2007, pp. 36–45.
- [50] M. Agosti, N. Ferro, Adding Advanced Annotation Functionalities to an Existing Digital Library, in: A. D’Atri, M. De Marco, N. Casalino (Eds.), Interdisciplinary Aspects of Information Systems Studies, Physica-Verlag, Heidelberg, Germany, 2008, pp. 279–286.
- [51] M. Agosti, N. Ferro, Annotations as Context for Searching Documents, in: F. Crestani, I. Ruthven (Eds.), Proc. 5th International Conference on Conceptions of Library and Information Science – Context: nature, impact and role (CoLIS 5), Lecture Notes in Computer Science (LNCS) 3507, Springer, Heidelberg, Germany, 2005, pp. 155–170.
- [52] M. Agosti, N. Ferro, Search Strategies for Finding Annotations and Annotated Documents: the FAST Service, in: H. Legind Larsen, G. Pasi, D. Ortiz-Arroyo, T. Andreasen, H. Christiansen (Eds.), Proc. 7th International Conference on Flexible Query Answering Systems (FQAS 2006), Lecture Notes in Artificial Intelligence (LNAI) 4027, Springer, Heidelberg, Germany, 2006, pp. 270–281.
- [53] N. Ferro, Annotation Search: The FAST Way, in: M. Agosti, J. Borbinha, S. Kapidakis, C. Papatheodorou, G. Tsakonas (Eds.), Proc. 13th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2009), Lecture Notes in Computer Science (LNCS) 5714, Springer, Heidelberg, Germany, 2009, pp. 15–26.
- [54] M. Agosti, O. Conlan, N. Ferro, C. Hampson, G. Munnely, Interacting with Digital Cultural Heritage Collections via Annotations: The CULTURA Approach, in: S. Marinai, K. Marriot (Eds.), Proc. 13th ACM Symposium on Document Engineering (DocEng 2013), ACM Press, New York, USA, 2013, pp. 13–22.
- [55] W3C, RDF Schema 1.1 – W3C Recommendation 25 February 2014, <http://www.w3.org/TR/rdf-schema/> (February 2014).
- [56] ISO/IEC 2382:2015, Information technology – Vocabulary, Recommendation ISO/IEC 2382:2015 (2015).