eFRBR: An entity model for FRBR

Vincenzo Maltese, Fausto Giunchiglia, Amit Sarangi, Stella Margonar
DISI – University of Trento, Italy

Abstract. Existing tools for library search typically support queries that directly exploit document properties such as title, author and subject. A lot of work in Knowledge Organization (KO) and Knowledge Representation (KR) shows that there is a need for more expressive queries where it is possible to predicate about the properties of relevant entities and, in particular, of those which are mentioned in the subject. These entities include people, organizations, locations, and events. Our proposed solution consists in the definition and exploitation of an *entity model* for FRBR, called *eFRBR*, which captures the main entities and properties informally defined in FRBR. When a specific use case needs to be addressed, eFRBR allows for the construction of a *knowledge graph* which codifies all the relevant entities which can then be suitably queried to obtain the desired results. We prove the validity of our approach in a use case addressing the scientific production at the University of Trento in Italy.

1. Introduction

So far, Library and Information Science has approached document management through the adoption of KO principles and techniques. Documents are classified, indexed and searched via properties such as *title*, *author* and *subject*. Controlled vocabularies and query expansion mechanisms ensure high precision and recall in search. In our previous work [1] we illustrated how such techniques, despite being widespread and successful, are still limited in the expressivity of the queries that they support. One example of query which is not supported is:

(*) Give me documents *about* the longest river in US

The need for such kinds of queries, based on the properties of the entities which are subject of a document or are related to a document in other ways, has been recognized in KO (for instance because of the difficulties faced by end users in libraries [4]), in databases (think for instance to the success of SQL [5]) and KR (the usage of SPARQL to query RDF stores [6]) communities.

In this paper we move one step forward the vision presented in [1] and we show how for these queries to be supported in practice we need to develop an *entity model* based on KR principles which provides the relevant *entity types* as well as the *terminology* which is needed to express them. The starting point in the definition of the entity model is FRBR [3]. Three features make FRBR an excellent starting point for our work (see also Tillett [11] and Coyle [12]):

- it provides a *conceptual* framework that can guide in the development of concrete models to be adopted in bibliographic settings;
- being based on the entity-relationship paradigm, it allows identifying the most important entities and properties than need to be considered;
- it helps in characterizing documents taking into account their abstract (work and expression) and physical (manifestation and item) aspects.

Yet, FRBR does not fully comply with our desiderata as (see also Coyle [12]):

- it is agnostic to queries;
- it mostly concentrates on documents with little emphasis on the properties of the other entities, including those which can be mentioned in the subject of documents, such as persons, corporate bodies, locations and events;
- being a conceptual framework, it cannot be directly employed to support the envisioned queries, but it needs to be evolved into a *logical* model (where actual data elements are specified) and then implemented as a *physical* model (where specific representation languages and technologies are used, tailored to the specific use case they need to serve).

In addition, a major limitation of the existing implementations of FRBR stands in the fact that they tend to take it too literally and offer a straight one-to-one implementation of its four conceptual levels (work, expression, manifestation, item) without considering the ontological properties of entities and, therefore, the way in which they will be instantiated and queried.

In this paper we present the *logical* entity model for FRBR, that we call ***eFRBR*** (for "*entity model for FRBR*"), and we show how it can be customized into a *physical* model tailored for a specific use case. The relevant entity types are defined building upon the approach presented in [2]. The terminology which is needed to express the entity properties is defined using the DERA methodology [1]. Properties include the class of entities (e.g. *city*), their relations with other entities (e.g. Trento is *part of* Italy), and their attributes (e.g. the *altitude* of Trento is 194 m). When a specific use case needs to be addressed we first customize the model and the terminology, and then we build a *knowledge graph* that is used to answer queries such as (*). A knowledge graph is a graph of entities which are instances of the entity types, defined using the terminology, and interconnected between them via relations. The knowledge graph is constructed by integrating information from multiple sources, including selected authority files.

The rest of the paper is organized as follows. In Section 2 we briefly describe FRBR. In Section 3 we introduce the eFRBR model and compare it with FRBR by spotting similarities and differences. In Section 4 we describe the eFRBR terminology. In Section 5 we proceed by giving a formalization of the kind of queries that can be supported by eFRBR and its terminology. In Section 6 we present the University of Trento use case and describe in particular the *physical* model we implemented (derived from the logical model), the difficulties faced in developing the knowledge graph, and the user interface of a search facility we developed to answer the queries. Section 7 concludes the paper by summarizing the work done and the next steps.

2. The FRBR Model

The Functional Requirements for Bibliographic Records (FRBR) entity-relationship reference model [3] was created by the International Federation of Library Associations and Institutions (IFLA). FRBR lays down the way in which bibliographic records and related entities should be described, and the standard user tasks exploiting them. The focus of FRBR is on those entities which are usually catalogued in authority records. As shown in Figure 1, it arranges entities in three groups. Group 1 entities are the products of intellectual or artistic endeavor. They include work, expression, manifestation and item. Group 2 entities are the entities responsible for the creation, realization, production, dissemination and custodianship of the group 1 entities, and include persons and corporate bodies. Group 3 entities serve as subjects of the group 1 entities. They include concept, object, event and place. All the entities of group 1 and 2 can also serve as subjects of the group 1 entities.

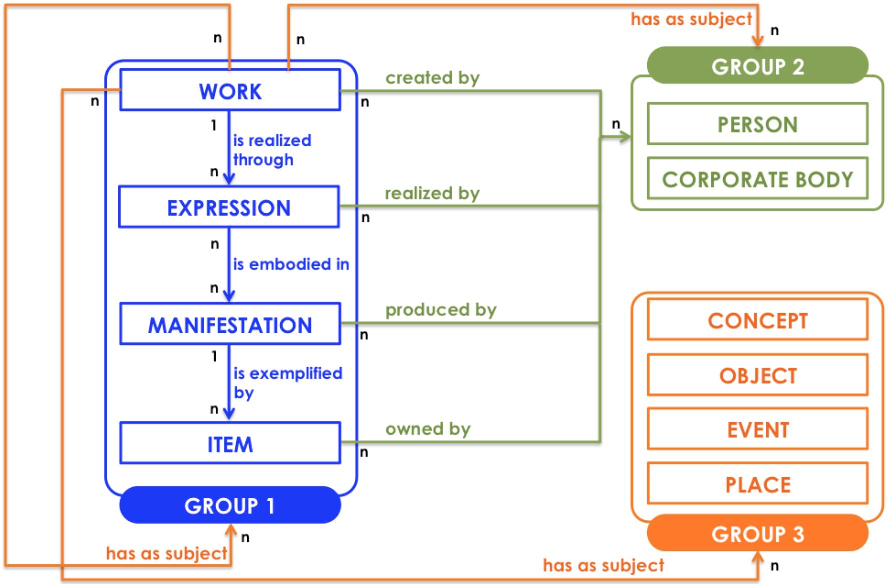


Figure 1 - The FRBR model

Within group 1, each *work* denotes a distinct intellectual or artistic creation (the original idea). An *expression* denotes the form in which a certain work is realized. A *manifestation* constitutes the embodiment of an expression in some format or medium. An *item* is an exemplar of a manifestation. Work and expression pretty much reflect the content, i.e. the abstract aspect of an information resource. Manifestation and item reflect the shape through which the content is experienced, i.e. the physical aspect of an information resource. A work (e.g. a poem) can be realized in multiple expressions (e.g. as text or sound), in turn made concrete via multiple manifestations (e.g. as paper or audio tape) and items (the multiple copies produced). Manifestation captures the invariants across all the items. In case of artistic creations there might be just one item for a certain work (e.g. in case of a statue). When multiple copies are possible there is typically an entity responsible for their production and dissemination (e.g. a publisher, a producer or a distributor).

Relations between works indicate the various ways in which a work can be considered equivalent to another work (*equivalent work*), derived from another one (*derivative work*), or can be the subject of another one (*descriptive work*) [11]. More in detail, a work is considered equivalent to another work when no significant intellectual or creative effort is taken to produce the new work from the original work (e.g. a copy of the original work); a work is derived from another work when significant intellectual or creative effort is taken and the modifications are judged as substantial (e.g. a summary of the original work); a work is descriptive of another work when it is based on the original work (e.g. a commented version or a critique of a work). Relations between expressions indicate variations in the form. Relations between manifestations indicate variations in the production and distribution of the various copies in various formats or media. Expressions, manifestations and items corresponding to the same work form a family of creations.

The entities in the FRBR model are all associated with the specifications of relevant attributes at the conceptual level [11], i.e. they are not specified in terms of actual data elements but they are rather described in terms of certain basic characteristics that need to be captured by any concrete representation of those entities. For instance, the "statement of responsibility" of a manifestation may be represented as a set of attributes describing the persons or organizations responsible for the production and distribution of the work or in alternative it may be represented as a bunch of relations linking to corresponding representations of the persons and corporate bodies.

3. The eFRBR model

Figure 2 shows the entity types of the *logical* entity model for eFRBR. The complete list of entity types and their properties (attributes and relations) is given in the appendix with a mapping to both FRBR and Dublin Core. In the picture, colors are used to show the mapping between the entity types in the schema and the entity types in FRBR. In particular, Work and Product are represented in blue and correspond to group 1 entities; Agent, Organization and Person are represented in green and correspond to group 2 entities; Location, Event, Entity (what in FRBR is called Object) and Concept are represented in orange and correspond to group 3 entities. Thing is introduced by us to allow a subject to be an Entity or a Concept; Agent generalizes Organization and Person.

Though FRBR is broader in scope, for the purposes of this work (i.e. supporting library search) we extended the general entity model to *literary works*. The dashed red line separates the general model from the extension of the model in this domain where Literary Work extends Work, and Publication extends Product. In any case, the model can be extended, both in terms of entity types and properties, whenever new requirements emerge. For instance, works relevant in other domains may include Cinematographic Work (which manifests into movies), Musical Work (which manifests into songs) and Visual Art (which manifests into paintings).

The most important relations include:

- ***is-a:*** it is the relation that enables inheritance; it indicates a rigid specialization, i.e. properties of the parent entity type are always inherited by the child entity type (sub-type). The latter extends the parent entity type by exposing additional properties. For instance, in the Literary Work entity type the property *creator* is inherited from the Work entity type, while the property *language* is defined at the level of Literary Work. In fact, *language* does not make sense for all Works (e.g. it does not for statues).
- ***part-of:*** it connects a part with the whole; it corresponds to the "whole/part" FRBR relation.
- ***instance-of:*** it connects an instance of any entity type more specific than Entity to the corresponding Concept denoting its entity class; for example, Mississippi is an instance of the entity type Location and may be defined as being of class river. Notice that in our approach we allow exactly one instance-of relation per entity.
- ***manifestation-of:*** it indicates the various ways in which a Work gets manifested in some physical form such that can be experienced by people; it corresponds to the "is embodied in" FRBR relation defined between an expression and a manifestation.

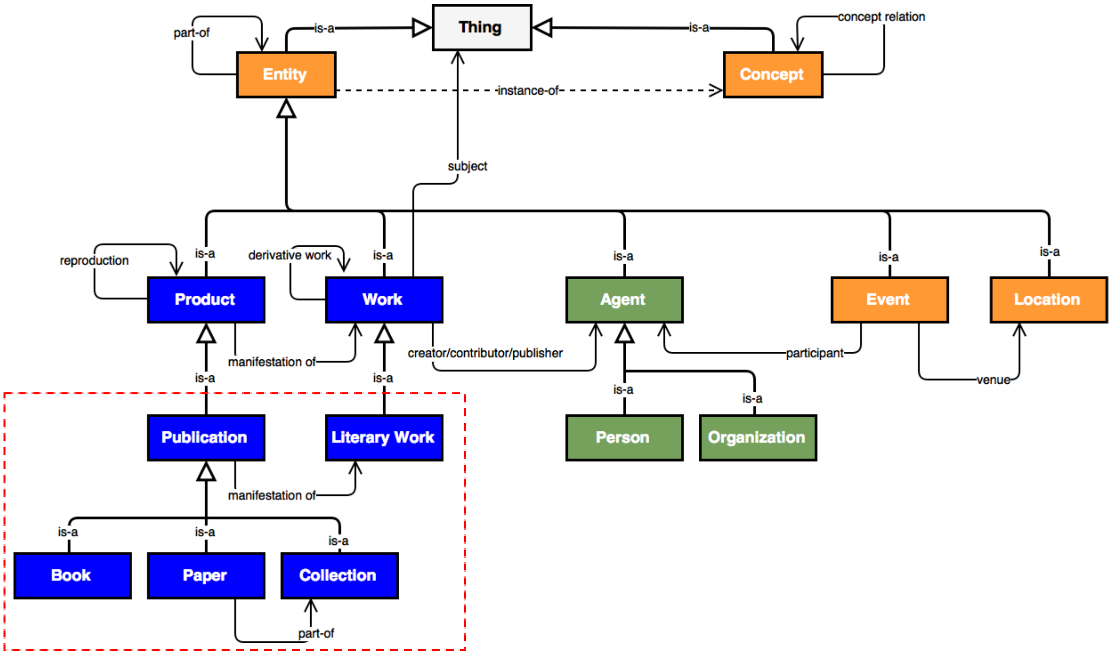
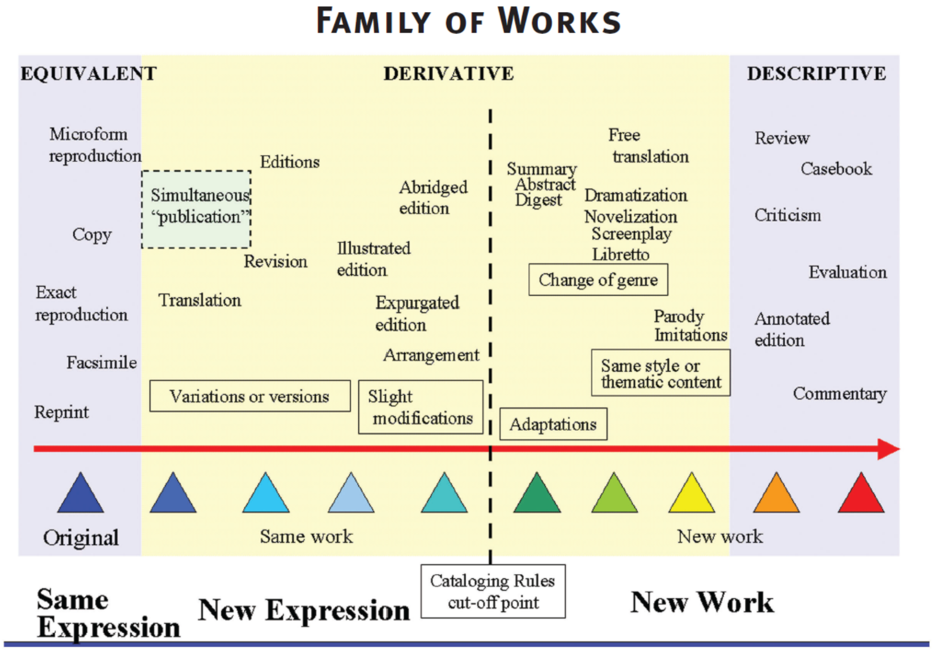


Figure 2 – The entity types in the schema of the bibliographic domain

In addition, we defined the relations necessary to codify those in Figure 3 and taken from the work by Tillet [11]. She observed that there is a continuum of derivations that, starting from an original work, can generate equivalent works (e.g., reprint), can generate new works (e.g., summary) or describe existing works (e.g., review). Consistently, we defined the following relations:

- ***derivative work***: it indicates the various ways in which a new Work originates from another Work. It captures derivative relations in Figure 3; in fact *derivative work* is the most general of those relations. The relation is defined between Works because related entities differ in content.
- ***reproduction:*** it indicates the various ways in which a Product can be identical to another Product. It captures equivalent relations in Figure 3; we consider *reproduction* to be the most general of those relations. Notice that this relation is defined between Products because related entities do not differ in content but in format.
- ***subject:*** it associates a Work to a Thing, that in turn can be any Concept (e.g. river) or Entity (e.g. Mississippi). In capturing descriptive relations in Figure 3, it can also indicate the various ways in which a Work describes another Work. For instance a review of the book entitled "The Adventures of Huckleberry Finn" would have both "review" (a concept) and the original work (an entity) as subject.



Relationships in the Organization of Knowledge, edited by Carol A. Bean and Rebecca Green, 2001, p. 23, "Bibliographic Relationships" by Barbara B. Tillett, Figure 2, © 2001 Kluwer Academic Publishers Boston, with kind permission of Kluwer Academic Publishers.

Figure 3 - The continuum from the same work to new works (taken from [11])

Figure 4 exemplifies the differences between FRBR group 1 entities and the elements - entity types and corresponding instances - of eFRBR. The latter is derived from FRBR by following the standard way by which data modeling is addressed in KR:

1. It enforces the separation between the entity types on the one side (called the TBox) and their instantiation on the other side (called the ABox) [10]. The entity types of eFRBR (what we call the *schema*) are kept separate from their instances (what we call the *knowledge graph*).

2. It captures the nature of the various real world entities to be modeled. FRBR group 1 entities capture ontologically distinct aspects of intellectual creations. *Work* and *Expression* are meant to capture the abstract aspect, i.e. the idea, with the latter specifying the various ways in which the same (original) idea can be expressed in various forms or can generate new (derived) ideas. Expression is therefore relational in nature. *Manifestation* is meant to capture the physical aspect, i.e. the various ways in which a certain idea can take a form that can be experienced concretely by people. Finally, *Item* captures the actual physical instance. For this reason, we define only two top level *entity types*: ***Work*** describes the abstract aspect of intellectual creations (it captures FRBR *work*); ***Product*** describes their physical aspect (it captures FRBR *manifestation*). The fact that a Product can be tangible (e.g., a volume) or virtual (e.g., a computer file) is described via the introduction of specific properties (e.g., weight in the first case and size in bytes in the second case). The ***derivative work*** relation between Works describes the various ways in which Works can be generated from one other

(it captures FRBR *expression*) in arbitrarily long chains (e.g., a summary of a translation of a certain work). In a chain of such relations original ideas can be tracked following the relations backwards. The ***manifestation-of*** relation allows a Product to the associated to the corresponding Work. These two entity types are taken separate from their abstract and physical ***instances*** which constitute the content of the knowledge graph. The abstract instances are instances of Work, while the physical instances are instances of Product (the latter captures FRBR *item* while the abstract instances are not explicitly distinguished from their entity types in FRBR). Any new FRBR expression generates a new abstract instance of Work connected to the original work via a relation more specific than derivative work.

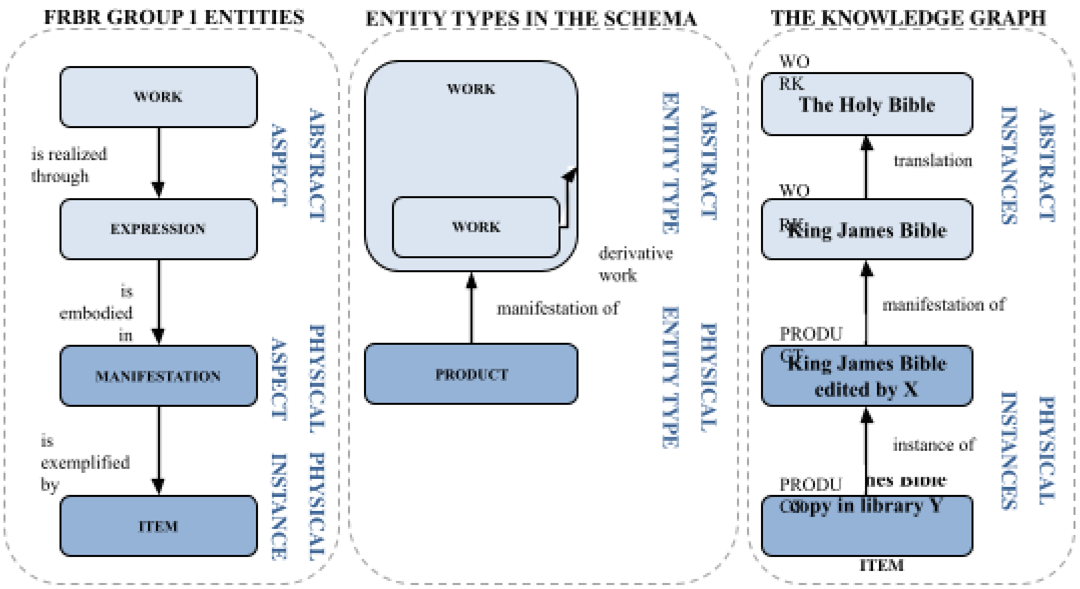


Figure 4 - Differences between our entity types and FRBR group 1 entities

3. It supports the definition of sub-types, thus supporting the propagation of properties by ontological similarity between entities. The top level entity types Work and Product can be specialized into sub-types as required by the specific use case. For instance, in Figure 2 the entity type Work is already specialized into Literary Work, while the entity type Product is specialized into Publication, in turn further specialized into Book, Paper and Collection.

4. It supports the specification of properties and sub-properties at arbitrary levels of specificity. Following the continuum depicted in Figure 3, the *derivative work* relation can be specialized into more specific strong relations (those on the right side of the figure), i.e. they link two different intellectual creations (e.g., summary), or weak relations (those on the left side of the figure), i.e. they link different expressions of the same intellectual creation (e.g., translation). Clearly this applies to relations and attributes of any kind. For instance, we can define *author*, *artist* and *composer* as specializations of *creator*.

As a comprehensive example, in Figure 4 we provide four instances. “The Holy Bible” and “The King James Bible” are instances of (Literary) Work. The latter is a *translation* (a relation more specific than derivative work) in English of the former. “The King James Bible edited by X” is an instance of a sub-type of Product that is “Book edited by X”. Notice that the entity types of the latter kind are special in that some of the attributes and relations of their instances are constrained in their values, while they can differ in others. For instance, all the instances of “Book edited by X” will have title “King James Bible” and editor “X”, but they can differ in their collocation in a library. One of its copies is stored in library Y.

Such ontological distinctions make our formalization very different from RDA[1] which provides a straight one-to-one mapping with FRBR. It is instead quite similar to BIBFRAME where FRBR objects are mapped into two entity types, one abstract (the Work) and one physical (the Product)[2]. Still, in our model the physical entity is a manifestation (and not an instance) of the abstract entity.

4. The eFRBR terminology

We employ the *DERA methodology* [1] to develop the terminology which is needed to describe the entity properties of the model. The methodology is inspired to the *analytico-synthetic approach* [8] that is known to guarantee the development of high quality vocabularies [9].

As it is shown in the example in Figure 5, the terminology developed using DERA takes a form similar to a thesaurus, as described in particular in the ISO 25964-1 standard [7], in that concepts (a) are connected via relations, (b) are expressed in natural language by terms, and (c) form hierarchies that are grouped into categories. In particular, each entity type generates a hierarchy of category ENTITY, each relation generates a hierarchy of category RELATION and each attribute generates a hierarchy of category ATTRIBUTE. More in detail, to describe documents:

- The category ENTITY includes a hierarchy rooted in *work* and one rooted in *product* to capture the various ways in which the two top level entity types of eFRBR can be specialized (see also item 3 in the previous section).
- The category RELATION includes the hierarchies of the most common relation names and capture the various ways in which relations can be specialized (see also item 4 in the previous section). In particular, the hierarchy rooted in *derivative work* provides (some of) the derivative relations depicted in Figure 3, while the hierarchy rooted in *reproduction* provides (some of) the equivalent relations depicted in Figure 3.
- Similarly, the category ATTRIBUTE includes the hierarchies of the most common attribute names and corresponding values. For instance, *ISBN* is an identifier used for books; *weekly* and *annual* are two possible values for *frequency* (of publication).

When instances of a certain entity type are defined, the terminology allows expressing that for instance a certain work is actually a literary work (concept #3) which is a translation (concept #17 which is more specific than derivative work) of another work and that it gets manifested as a chapter (concept #14) of a book (concept #12).

[1] http://www.rda-jsc.org/rda.html

[2] http://www.loc.gov/bibframe/docs/bibframe-profiles.html#examples

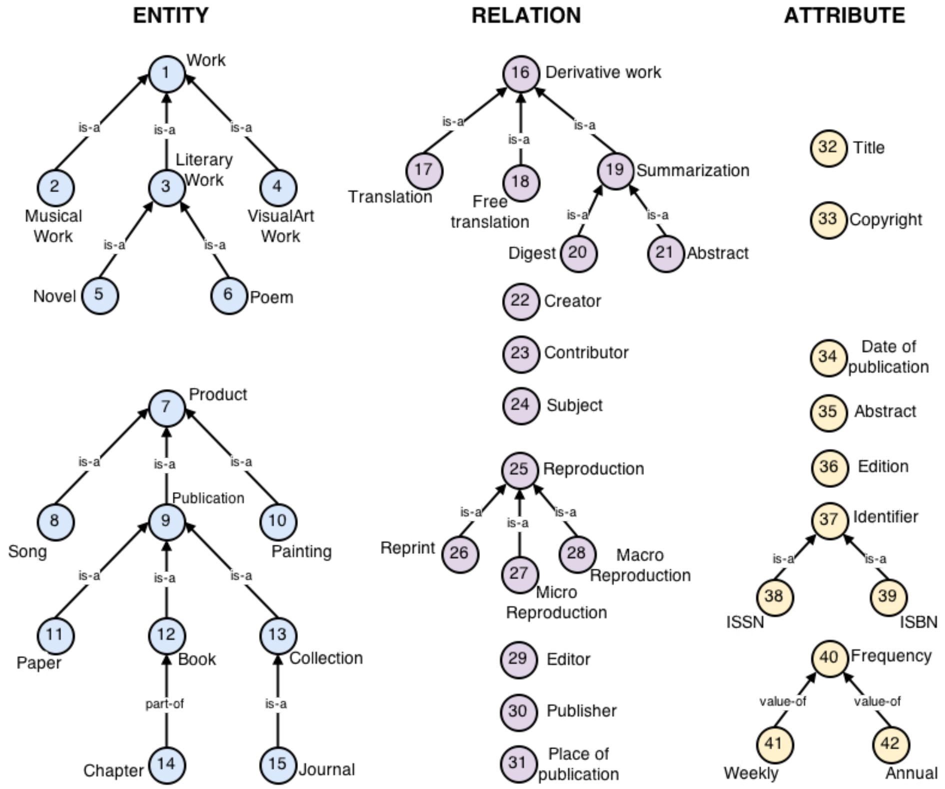


Figure 5 - Example of terminology developed using DERA

Notice that differently from a standard thesaurus, in our settings the semantic relations between concepts include *is-a* (between entity classes, and between attributes/relations names), *part-of* (between entity classes) and *value-of* (between an attribute name and corresponding values, thus specifying the range of possible values). As explained in [1, 13], these kinds of semantic relations are necessary to support automated reasoning.

5. Answering queries

The eFRBR entity model and the terminology we propose support a broad range of very expressive queries. For instance, concerning documents[3] they can support the following generalized query:

(**) Find documents where property x denotes an entity of type y with property z=$

[3] More in general eFRBR can support queries based on any kind of entity.

where x is any property of a document (e.g. subject, author, editor), y is any type of entity (e.g. person, organization, location, event or even another document) and z is any of the entity properties (e.g. for a person we might have date of birth, place of birth, nationality, affiliation).

In eFRBR queries about documents correspond to either a Work or Product (or any of their more specific concepts taken from respective hierarchies in the ENTITY category) according to whether the user is interested in their abstract or physical aspect, respectively; x corresponds to any relation of those entity types (or any of their more specific concepts taken from hierarchies in the RELATION category); y corresponds to any entity type in the eFRBR model; x corresponds to any relation or attribute of those entity types (or any of their more specific concepts taken from hierarchies in the RELATION or ATTRIBUTE categories); consistently, $ corresponds to a value (e.g. a concept which is a value-of an attribute in ATTRIBUTE categories). For instance:

> *type* = Work and *translation* = $1 and $1.*type* = Work and $1.*title* = "The Adventures of Huckleberry Finn"
> (find works that are translations of "The Adventures of Huckleberry Finn")
>
> *type* = Product and *subject* = $1 and $1. *type* = river and $1.*lenght* > 2000 and $1.*part-of* = "US"
> (find products about rivers in US which are longer than 2000 km)

In other words, the generalized query (**) covers all cases in which the user is looking for a document on the basis of the properties of the entities which are related in some way to such document.

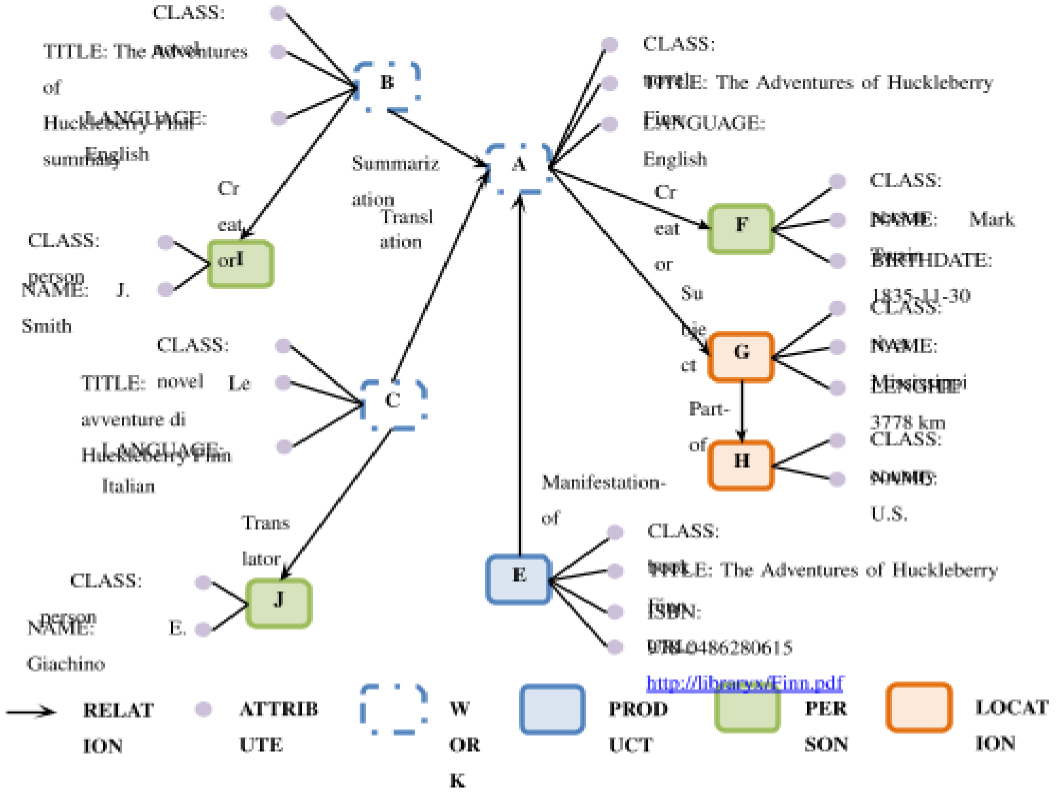


Figure 6 - Example of knowledge graph

Still, a fundamental step towards being able to answer such queries is to develop a knowledge graph which encodes the data that need to be queried in form of interconnected entities. The entity types and the terminology of eFRBR allow defining the relevant entities uniformly, regardless of their initial source. However, appropriate methodologies and tools need to be put in place to suitable integrate the various information sources, including for instance different institutional repositories and authority files, thus generating a knowledge graph that represents the whole knowledge of the institution. Once the knowledge graph is constructed each query corresponds to exploring the graph and it is implemented concretely as a logical reasoning task.

Figure 6 provides an example of knowledge graph where the entities are described according to the entity types in Figure 2 and (an extension of) the terminology in Figure 5. In fact, all the attribute and relation names, as well as the values of the attributes of data type concept (i.e. all those that can be controlled), are taken from hierarchies of corresponding categories. For instance, the term “title” appearing as attribute name corresponds to concept #32 in the ATTRIBUTE category; the term “creator” appearing as relation name corresponds to concept #22 in the RELATION category; the term “novel” appearing as the value of the class attribute corresponds to concept #5 in the ENTITY category. Class is a special attribute that implements the instance-of relation.

In the following we give examples of queries of kind (**) formulated using the terminology in Figure 5 and answered using the knowledge graph in Figure 6 and that capture the four aspects of intellectual creations addressed by the FRBR model:

Find works (the user is interested in the abstract aspect):

> *type* = Work and *creator* = $1 and $1.*type* = person and $1.*birthdate* = "1835-11-30"
> (find any work whose creator is born on 1835-11-30)
>
> The query will return the entity labelled A (an instance of Work) because its creator Mark Twain is born on 1835-11-30. The query also shows the ability to navigate the knowledge graph, thus querying by the properties of the entities which are related to the target entity.

Find expressions (the user is interested in the abstract aspect):

> *type* = Work and *derivative-work* = $1 and $1.*type* = Work and $1.*title* = "The Adventures of Huckleberry Finn"
> (find any derivative work of the original work "The Adventures of Huckleberry Finn")
>
> The query will return the entities labelled B and C (two instances of Work) because the concepts Summarization and Translation are both more specific than derivative work in the corresponding hierarchy in Figure 5. The query shows the ability to reason about the concepts of the relation (or attribute) names.

Find manifestations (the user is interested in the physical aspect):

> *type* = Product and *ISBN* = 978-0486280615
> (find any product with *ISBN* equal 978-0486280615)
>
> The query will obviously return the entity labelled E (an instance of Product).

Find items (the user is interested in the physical aspect):

> *type* = Product and *manifestation-of* = $1 and $1.*type* = Work and $1.*subject* = $2 and $2.*class* = river and $2.*part-of* = "US"
> (find all the items corresponding to works about rivers in US)
>
> The query will obviously return the entity labelled E (an instance of Product). The query also shows the ability to navigate the graph at arbitrary depth by nesting properties.

Notice that the difference between Product and its instances is that some of the attribute values of the instances are fixed by the Product (i.e. they are the same in all instances), for instance the ISBN or the editor, and others depend on the actual instance, for instance the URL of the file or the collocation of the physical volume on the shelves.

6. The University of Trento use case

We validated the proposed approach in the context of a project, called *Digital University* (started in October 2014) whose broader aim is to face the sparse and heterogeneous nature of the information sources and offer powerful search and analytics facilities[4] over the knowledge assets of the University of Trento. The use case shows the validity of our proposed approach in that:

1. A *physical* model is built by appropriately customizing the entity types and the terminology of eFRBR;
2. A knowledge graph following the entity types and the terminology above has been developed by integrating the various information sources available;
3. Powerful search services exploiting the knowledge graph have been developed..

The IT infrastructure at support of the data representation, the definition of the terminology, the data integration and the exploitation of the knowledge graph is based on a KR framework, called SCORE, developed by the KnowDive research group at the University of Trento.

The information sources we selected contain metadata about several entity types: ADA contains information about people and institutions; TIMESHEET contains projects information; ESSE3 contains teaching information (e.g. courses and exams); the remaining datasets represent various collections of research products such as papers, books, thesis and patents. They partially overlap in metadata and entities described. One of those datasets called IRIS[5] is based on DSpace[6] open repository software, two of them are based on EPrints[7] and the remaining one, called TEA, is proprietary software developed to store doctoral and master thesis. IRIS is the main institutional document repository. It contains information about around 102,000 research products, and related organizations (e.g. publishers) and events (e.g. the conferences where papers were presented).

ADA is the system that at University of Trento acts as central name authority for people. It contains information about around 127,000 affiliates to the University of Trento. Thanks to ADA, name authority control is enforced in TIMESHEET, ESSE3 and partially in IRIS, given that it supports only the recognition of internal people. Authority control is totally absent in the other datasets. No name authority is enforced for other kinds of entities. This basically means that authority files are not exploited at all. As initial step to compensate for this limitation, we identified an authority for location names. Given that we developed it, we took the GeoWordNet [14] geo-spatial resource that contains, among other things, information about 23,000 cities and 350 countries.

6.1 Customizing the Entity Types and the Terminology

The customization is the result of an activity of data analysis aimed at checking the coverage of the model in terms of entity types, properties and terminology against the data from the various data sources as well as the use case requirements, in terms of services to be provided to the end user. The final result is the actual *physical* model and related terminology that serves the various applications.

Given that all the research products contained in the various document repositories are in digital form, the *physical* model we implemented is a simplification of the *logical* model presented in Section 3. In the physical model the abstract aspect of the research products is fused with the physical aspect and therefore research products are modeled as instances of one entity type only called Product. In fact,

[4] In this paper we focus only on the search facility and in particular on its ability to search for documents.

[5] https://wiki.u-gov.it/confluence/pages/releaseview.action?pageId=51810588

[6] http://www.dspace.org/

[7] http://www.eprints.org/uk/

we assume that users are only interested to search for the virtual items of works. This is consistent with Coyle's observation [12] that the four levels of FRBR group 1 entities should be seen as aspects of the same entity (rather than four different entities) and that the way in which they should be represented as metadata depends on the use cases.

Deciding about the granularity and the level of specificity of the entity types in terms of sub-types and related properties to be defined is also a function of the use case. We tried to be minimalistic and avoided to introduce new sub-types unless strictly necessary. In the IRIS repository each research product is associated a type taken from the Italian Ministry of Research (MIUR) categorization scheme, which is very fine-grained. For instance, the category *"Translation of a Book"* can be captured by using the existing entity type Book and by specifying that it is a derivative work via an appropriate *translation* relation with the original book which can be specified in IRIS[8]. As further example, though in IRIS we can identify journal papers, conference papers and poster papers, no additional attributes are foreseen for them w.r.t. those already foreseen for papers. Therefore they can be all defined as instances of Paper and further distinguished by setting up the class attribute with the appropriate value taken from the ENTITY hierarchy in Figure 5. Thanks to this analysis we decided to introduce only the new entity type Patent as sub-type of Product.

Finally, the terminology is often extended with more specific relations and attributes as a function of the use case. This is done in order to cope with concepts and terms emerging from the data. This is pretty similar to the well-known literary warrant principle. In particular, new relations such as Translator, Reviewer, and Patentee were added in the RELATION category; the attribute Identifier in the ATTRIBUTE category were further specialized into URL, DOI, ISBN, ISSN, and ISMN.

6.2 Building the Knowledge Graph

By applying data integration techniques supported by SCORE (which are out of the scope of this paper), metadata about entities were extracted from the various information sources. By applying some transformations and Natural Language Processing techniques, the extracted entities were encoded following the customized entity types and terminology above, thus enforcing vocabulary control, and used to populate the knowledge graph. Relations between entities were created whenever possible, thus enforcing name authority as much as possible. For instance, each publication extracted from IRIS was associated to corresponding authors and contributors extracted from ADA; each mention to a location was linked to the corresponding city or country taken from GeoWordNet. Notice that the knowledge graph constitutes a new information layer built on top of the existing information resources, which remain therefore unchanged.

Unfortunately, we found out that research products are poorly indexed in the source repositories. For instance, in IRIS subjects are absent, while keywords are associated only to 8% of the research products. Keywords can be in any language. We took 5000 random products from IRIS and manually annotated keywords and abstracts with corresponding entities whenever mentioned. In this way we

[8] This decomposition supports the identification of the basic building blocks that, by following the Meccano property described by Ranganathan, can be used to construct any combination on demand. For instance, we can represent the translation of a review of a book chapter, even though no category is foreseen by MIUR. This has concrete advantages when data coming from different sources gets integrated as at their origin data may follow different organizational principles.

identified 198 persons (e.g. Aristotle), 171 locations (e.g. Pasadena), 87 organizations (e.g. University of Cambridge), 26 events (e.g. Trento's council), and 88 works (e.g. "Critique of Pure Reason" by Emmanuel Kant). We also identified 18 entities of a kind that was not considered in eFRBR and we therefore categorized them as generic instances of Entity (e.g. Canopus the star). The manually recognized entities were then encoded following eFRBR and integrated in the knowledge graph.

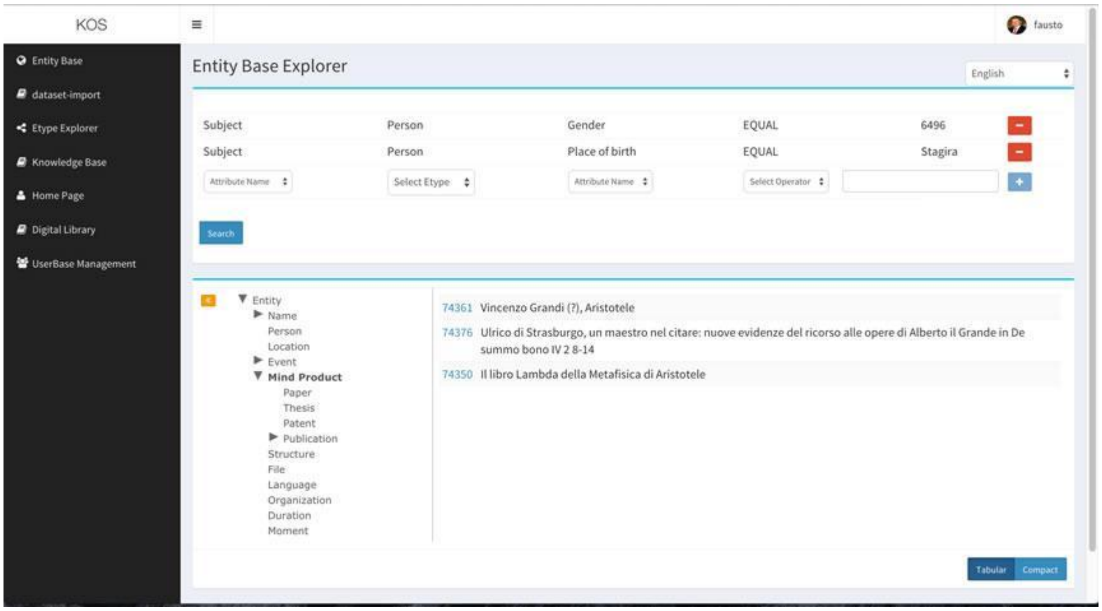
This sample was used to train a Named Entity Disambiguation (NED) tool that allows recognizing entities from keywords and abstracts of research products. Whenever a match was found, they were automatically mapped with entities in the knowledge graph. Concepts mentioned in keywords and abstracts were automatically identified and disambiguated by a Word Sense Disambiguation (WSD) tool and mapped with the terminology. In other words, the NED automates name authority, while the WSD automates vocabulary control, thus compensating for the lack of human support.

6.3 Answering queries

Figure 7 shows the user interface (UI) of the search facility we developed to query the knowledge graph. It allows searching for entities of various types in very expressive ways.

The entity type is selected from the tree rooted in Entity and search criteria can be specified accordingly. In the picture, and with reference to the generalized query (**) given in Section 5, the first occurrence of *"Attribute Name"* corresponds to the parameter x, *"Etype"* corresponds to y, the second occurrence of *"Attribute Name"* corresponds to the parameter z and the empty field before the + button allows specifying the parameter $.

For instance, here Mind Product (that is a synonym of Product) is selected. Once the entity type is chosen the user can select one or more of its properties from the "Attribute Name" dropdown menu. The available names here are those of the properties defined for the selected entity type. Here for instance subject is selected twice. Given that subject also admits entities as values (i.e. it is relational), the user can select that she is looking for products whose subject is a person of gender equal "male" (the concept 6496 stands for "male" and pops up when "male" is typed), and place of birth equal "Stagira". The query returns three products which are about Aristotle (who is a male born in Stagira). Figure 8 shows the attributes of the second product in the result set, which is a book chapter.



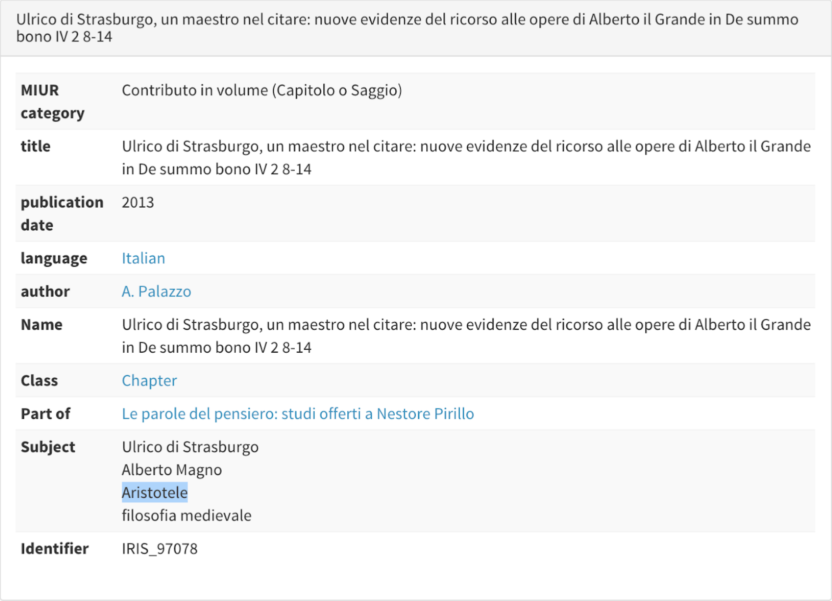
Subject	Person	Gender	EQUAL	6496
Subject	Person	Place of birth	EQUAL	Stagira

74361 Vincenzo Grandi (?), Aristotele

74376 Ulrico di Strasburgo, un maestro nel citare: nuove evidenze del ricorso alle opere di Alberto il Grande in De summo bono IV 2 8-14

74350 Il libro Lambda della Metafisica di Aristotele

Figure 7 - The UI to issue the search queries: example of query and corresponding result



Ulrico di Strasburgo, un maestro nel citare: nuove evidenze del ricorso alle opere di Alberto il Grande in De summo bono IV 2 8-14

MIUR category	Contributo in volume (Capitolo o Saggio)
title	Ulrico di Strasburgo, un maestro nel citare: nuove evidenze del ricorso alle opere di Alberto il Grande in De summo bono IV 2 8-14
publication date	2013
language	Italian
author	A. Palazzo
Name	Ulrico di Strasburgo, un maestro nel citare: nuove evidenze del ricorso alle opere di Alberto il Grande in De summo bono IV 2 8-14
Class	Chapter
Part of	Le parole del pensiero: studi offerti a Nestore Pirillo
Subject	Ulrico di Strasburgo Alberto Magno Aristotele filosofia medievale
Identifier	IRIS_97078

Figure 8 - The UI to issue the search queries: opening one of the results

7. Conclusions

In this paper we presented eFRBR, an ontologically sound entity model that supports a wide range of very expressive queries. The entity model is based on FRBR, thus formally representing creative works and related entities such as people, organizations, locations and events. The terminology which is needed to express the entity properties is defined using the DERA methodology. Queries formulated using the terminology are answered by exploiting a knowledge graph that instantiates the model. The knowledge graph is constructed by integrating relevant information from multiple sources. We demonstrated the validity of the approach in a concrete use case addressing the scientific production of the University of Trento in Italy.

Our future work includes the finalization of the proposed *logical* model with the definition of appropriate projection operations and a methodology which guide in the generation of the *physical* models which are necessary to address specific use cases. Such operations will specify if and how the different aspects of intellectual creations captured by eFRBR will have to be kept in the physical model, both in terms of entity types and properties. Moreover, we plan to formalize our methodology of data integration that guides in the generation of the knowledge graph from available information sources. We also plan to develop and commercialize a software solution embedding and extending these ideas to support institutions in managing and exploiting their key knowledge assets.

Acknowledgements

This work has received funding from the Digital University and the ITPAR (India-Trento Programme for Advanced Research) projects. We are grateful to Francesca Valentini, chief of the University of Trento research products archives, for her constant support. We also want to thank Amarsanaa Gambold and Enrico Bignotti for their early work on the theory of works, Gabriele Tonelli for the entity annotation and definition, Gabor Bella for the training of the NED and WSD tools, Ilya Zaihrayeu, Viktor Pravdin, Marco Marasca, Uladzimir Kharkevich, Francesco Bux and all the other members of the KnowDive group for their effort in the implementation of the KR framework SCORE that supports the definition, storage and search of entities. In addition, we would like to thank CINECA (http://www.cineca.it/) for their support on IRIS.

References

1. Giunchiglia, F., Dutta, B., Maltese, V. (2014). From Knowledge Organization to Knowledge Representation. Knowledge Organization Journal, 41(1), 44-56. Presented at ISKO UK 2013.
2. Giunchiglia, F., Maltese, V., Dutta, B. (2012). Domains and context: first steps towards managing diversity in knowledge. Journal of Web Semantics, 12-13, 53-63.
3. International Federation of Library Associations, & Institutions. Section on Cataloguing. Standing Committee (1997-2009). Functional Requirements for Bibliographic Records: final report. IFLA Study Group on the Functional Requirements for Bibliographic Records (Ed.). KG Saur Verlag Gmbh & Company.
4. Library of Congress (2007). Library of Congress Subject Headings: Pre- vs. Post-Coordination and Related Issues. Library of Congress; Cataloging Policy & Support Office; Technical Report.
5. Ramakrishnan, R., Gehrke, J. (2000). Database Management Systems. McGraw-Hill.
6. Prud’hommeaux, E., Seaborne, A. (2006). SPARQL Query Language for RDF. W3C Working Draft. http://www.w3.org/TR/2006/WD-rdf-sparql-query-20061004/

7. ISO 25964-1:2011. Information and documentation – Thesauri and interoperability with other vocabularies – PART 1: Thesauri for information retrieval.
8. Ranganathan, S. R. (1967). Prolegomena to library classification. London: Asia Pub. House.
9. Broughton, Vanda. 2006. The need for a faceted classification as the basis of all methods of information retrieval. Aslib Proceedings, 58(1/2): 49-72.
10. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (2002). The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press.
11. Tillett, B. (2005). What is FRBR? A conceptual model for the bibliographic universe. The Australian Library Journal, 54(1), 24-30.
12. Coyle, K. (2014). FRBR, Twenty Years On. Cataloging & Classification Quarterly, 1-21.
13. Maltese, V., Farazi, F. (2011). Towards the Integration of Knowledge Organization Systems with the Linked Data Cloud. UDC Seminar 2011.
14. Giunchiglia, F., Maltese, V., Farazi, F., Dutta, B. (2010). GeoWordNet: a Resource for Geo-Spatial Applications. 7th Extended Semantic Web Conference (ESWC). Lecture Notes in Computer Science, Springer-Verlag, Vol. 6088, pp 121-136.

Appendix: list of properties of the schema mapped with FRBR and Dublin Core

The table below provides the list of attributes and relations of the entity types in our schema. Each of them is associated a name, the data type, the mapping with the corresponding FRBR and Dublin Core (DC) property whenever applicable, and eventually some notes. Data types in angular brackets (< >) indicate entity types. Properties in light gray indicate those which are inherited from more general entity types.

Property name	Data type	FRBR	DC	Notes
THING				
Identifier	String			Identifier
CONCEPT				
Identifier	Long			It overrides THING.Identifier
Category	Char			One of E, R, A
SYNSET				
Identifier	Long			It overrides THING.Identifier
Language	String			ISO code of the language
POS	Char			One of n, a, v, r
Term	String [ ]	3.concept.term for the concept		
Gloss	String			
ENTITY				
Identifier	String			Inherited from THING
Name	String			
Class	<Concept>			Implements the instance-of relation
Start	Date			
End	Date			
Part of	<Entity> [ ]			
Description	String [ ]			
WORK				
Identifier	String	1.manifestation.manifestation identifier	Identifier	Inherited from THING
Name	String	1.work.title of the work	Title	Inherited from ENTITY
Class	<Concept>		Type	Inherited from ENTITY

Start	Date	1.work.date of the work	Date	Inherited from ENTITY
End	Date	1.work.intended termination	Date	Inherited from ENTITY
Part of	<Entity> [ ]		Relation	Inherited from ENTITY
Description	String [ ]	1.work.context for the work	Description	Inherited from ENTITY
Subject	<Thing> [ ]	1.work.subject relationships	Subject	
Audience	<Concept> [ ]	1.work.intended audience	Audience	
Creator	<Agent> [ ]	1.manifestation.statement of responsibility	Creator	
Contributor	<Agent> [ ]	1.manifestation.statement of responsibility	Contributor	
LITERARY WORK				
Identifier	String	1.manifestation.manifestation identifier	Identifier	Inherited from THING
Title	String	1.work.title of the work	Title	It overrides ENTITY.Name
Class	<Concept>		Type	Inherited from ENTITY
Start	Date	1.work.date of the work	Date	Inherited from ENTITY
End	Date	1.work.intended termination	Date	Inherited from ENTITY
Part of	<entity> [ ]		Relation	Inherited from ENTITY
Description	String [ ]	1.work.context for the work	Description	Inherited from ENTITY
Audience	<Concept> [ ]	1.work.intended audience	Audience	Inherited from WORK
Creator	<Agent> [ ]	1.manifestation.statement of responsibility	Creator	Inherited from WORK
Contributor	<Agent> [ ]	1.manifestation.statement of responsibility	Contributor	Inherited from WORK
Language	<Concept> [ ]	1.expression.language of expression	Language	
Copyright	String [ ]	1.expression.use restrictions on the expression	Rights	
PRODUCT				
Identifier	String	1.manifestation.manifestation identifier	Identifier	Inherited from THING
URL	String	1.manifestation.manifestation identifier	Identifier	It overrides THING.Identifier
Name	String	1.work.title of the work	Title	Inherited from ENTITY
Class	<Concept>		Type	Inherited from ENTITY
Start	Date	1.work.date of the work	Date	Inherited from ENTITY
End	Date	1.work.intended termination	Date	Inherited from ENTITY
Part of	<Entity> [ ]		Relation	Inherited from ENTITY
Description	String [ ]	1.work.context for the work	Description	Inherited from ENTITY
Audience	<Concept> [ ]	1.work.intended audience	Audience	Inherited from WORK
Creator	<Agent> [ ]	1.manifestation.statement of responsibility	Creator	Inherited from WORK
Contributor	<Agent> [ ]	1.manifestation.statement of responsibility	Contributor	Inherited from WORK
Producer	<Agent> [ ]	1.manifestation.fabricator/manufacturer		
Distributor	<Organization>[ ]	1.manifestation.publisher/distributor		
Format	<Concept>		Format	
Size	Long	1.manifestation.dimensions of the carrier		
PUBLICATION				
Identifier	String	1.manifestation.manifestation identifier	Identifier	Inherited from THING
Title	String	1.work.title of the work	Title	Inherited from LITERARY WORK
Class	<Concept>		Type	Inherited from ENTITY
Date of publication	Date	1.manifestation.date of publication/distribution	Date	It overrides ENTITY.Start
End	Date	1.work.intended termination	Date	Inherited from ENTITY
Part of	<Entity> [ ]		Relation	Inherited from ENTITY
Abstract	String [ ]	1.work.context for the work	Description	It overrides ENTITY.Description
Audience	<Concept> [ ]	1.work.intended audience	Audience	Inherited from WORK
Creator	<Agent> [ ]	1.manifestation.statement of responsibility	Creator	Inherited from WORK
Contributor	<Agent> [ ]	1.manifestation.statement of responsibility	Contributor	Inherited from WORK
Language	<Concept> [ ]	1.expression.language of expression	Language	Inherited from LITERARY WORK
Copyright	String [ ]	1.expression.use restrictions on the expression	Rights	Inherited from LITERARY WORK
Producer	<Agent> [ ]	1.manifestation.fabricator/manufacturer		Inherited from PRODUCT
Publisher	<Agent> [ ]	1.manifestation.publisher/distributor	Publisher	It overrides PRODUCT.Producer
Editor	<Organization>[ ]	1.manifestation.statement of responsibility	Contributor	
Price	Float			Inherited from PRODUCT

Edition	String	1.manifestation.edition/issue designation		
Place of publication	<Location>	1.manifestation.place of publication/distribution		
COLLECTION				
ISSN	String	1.manifestation.manifestation identifier	Identifier	It overrides THING.Identifier
Title	String	1.work.title of the work	Title	Inherited from LITERARY WORK
Class	<Concept>		Type	Inherited from ENTITY
Date of publication	Date	1.manifestation.date of publication/distribution	Date	Inherited from PUBLICATION
End	Date	1.work.intended termination	Date	Inherited from ENTITY
Part of	<Entity> [ ]		Relation	Inherited from ENTITY
Description	String [ ]	1.work.context for the work	Description	Inherited from ENTITY
Audience	<Concept> [ ]	1.work.intended audience	Audience	Inherited from WORK
Creator	<Agent> [ ]	1.manifestation.statement of responsibility	Creator	Inherited from WORK
Contributor	<Agent> [ ]	1.manifestation.statement of responsibility	Contributor	Inherited from WORK
Language	<Concept> [ ]	1.expression.language of expression	Language	Inherited from LITERARY WORK
Copyright	String [ ]	1.expression.use restrictions on the expression	Rights	Inherited from LITERARY WORK
Producer	<Agent> [ ]	1.manifestation.fabricator/manufacturer		Inherited from PRODUCT
Publisher	<Agent> [ ]	1.manifestation.publisher/distributor	Publisher	Inherited from PUBLICATION
Editor	<Organization>[ ]	1.manifestation.statement of responsibility	Contributor	Inherited from PUBLICATION
Price	Float			Inherited from PRODUCT
Edition	String	1.manifestation.edition/issue designation		Inherited from PUBLICATION
Place of publication	<Location>	1.manifestation.place of publication/distribution		Inherited from PUBLICATION
Volume	Integer	1.expression.sequencing pattern		
Number	Integer	1.expression.sequencing pattern		
Regularity	<Concept>	1.expression.expected regularity of issue		
Frequency	<Concept>	1.expression.expected frequency of issue		
PAPER				
ISBN	String	1.manifestation.manifestation identifier	Identifier	It overrides THING.Identifier
Title	String	1.work.title of the work	Title	Inherited from LITERARY WORK
Class	<Concept>		Type	Inherited from ENTITY
Date of publication	Date	1.manifestation.date of publication/distribution	Date	Inherited from PUBLICATION
End	Date	1.work.intended termination	Date	Inherited from ENTITY
Part of	<Entity> [ ]		Relation	Inherited from ENTITY
Description	String [ ]	1.work.context for the work	Description	Inherited from ENTITY
Audience	Concept [ ]	1.work.intended audience	Audience	Inherited from WORK
Creator	<Agent> [ ]	1.manifestation.statement of responsibility	Creator	Inherited from WORK
Contributor	<Agent> [ ]	1.manifestation.statement of responsibility	Contributor	Inherited from WORK
Language	<Concept> [ ]	1.expression.language of expression	Language	Inherited from LITERARY WORK
Copyright	String [ ]	1.expression.use restrictions on the expression	Rights	Inherited from LITERARY WORK
Producer	<Agent> [ ]	1.manifestation.fabricator/manufacturer		Inherited from PRODUCT
Publisher	<Agent> [ ]	1.manifestation.publisher/distributor	Publisher	Inherited from PUBLICATION
Editor	<Organization>[ ]	1.manifestation.statement of responsibility	Contributor	Inherited from PUBLICATION
Price	Float			Inherited from PRODUCT
Edition	String	1.manifestation.edition/issue designation		Inherited from PUBLICATION
Place of publication	<Location>	1.manifestation.place of publication/distribution		Inherited from PUBLICATION
Conference	<Event>			
AGENT				
Identifier	String			Inherited from THING
Name	String			Inherited from ENTITY
Class	<Concept>			Inherited from ENTITY

Start	Date			Inherited from ENTITY
End	Date			Inherited from ENTITY
Part of	<Entity> [ ]			Inherited from ENTITY
Description	String [ ]			Inherited from ENTITY
Position	<Location>			
PERSON				
Identifier	String			Inherited from THING
Name	String	2.person.name of person		Inherited from ENTITY
Class	<Concept>			Inherited from ENTITY
Date of birth	Date	2.person.dates of person		It overrides ENTITY.Start
Date of death	Date	2.person.dates of person		It overrides ENTITY.End
Affiliation	<Organization>[ ]	2.person.other designation associated with the person		It overrides ENTITY.Part of
Description	String [ ]			Inherited from ENTITY
Gender	Concept	2.person.other designation associated with the person		Possible values: {Male, Female}
Place of birth	<Location>	2.person.other designation associated with the person		It overrides AGENT.Position
Place of death	<Location>	2.person.other designation associated with the person		It overrides AGENT.Position
ORGANIZATION				
Identifier	String	1.manifestation.manifestation identifier		Inherited from THING
Name	String	2.corporate body.name of the corporate body		Inherited from ENTITY
Class	<Concept>			Inherited from ENTITY
Date of establishment	Date	2.corporate body.date associated with corporate body		It overrides ENTITY.Start
Date of disestablishment	Date	2.corporate body.date associated with corporate body		It overrides ENTITY.End
Part of	<Entity> [ ]			Inherited from ENTITY
Description	String [ ]	1.work.context for the work		Inherited from ENTITY
Seat	<Location>	2.corporate body.place associated with the corporate body		It overrides AGENT.Position
LOCATION				
Identifier	String			Inherited from THING
Name	String	3.place.term for the event		Inherited from ENTITY
Class	<Concept>			Inherited from ENTITY
Start	Date			Inherited from ENTITY
End	Date			Inherited from ENTITY
Part of	<Entity> [ ]			Inherited from ENTITY
Description	String [ ]			Inherited from ENTITY
Latitude	Float			
Longitude	Float			
Altitude	Float			
EVENT				
Identifier	String			Inherited from THING
Name	String	3.event.term for the event		Inherited from ENTITY
Class	<Concept>			Inherited from ENTITY
Start	Date			Inherited from ENTITY
End	Date			Inherited from ENTITY
Part of	<Entity> [ ]			Inherited from ENTITY
Description	String [ ]			Inherited from ENTITY
Venue	<Location> [ ]			
Participant	<Person> [ ]			