Semantic Web 0 (2011) 1
IOS Press

FRBR-ML: A FRBR-based Framework for Semantic Interoperability

Naimdjon Takhirov [a,*], Trond Aalberg [a], Fabien Duchateau [a,**], and Maja Žumer [b]
[a] *Department of Computer and Information Science*
Norwegian University of Science and Technology
NO-7491 Trondheim, Norway
E-mail: {takhirov,trondaal,fabiend}@idi.ntnu.no
[b] *University of Ljubljana, 1000 Ljubljana, Slovenia*
E-mail: maja.zumer@ff.uni-lj.si

Abstract. Metadata related to cultural items such as literature, music and movies is a valuable resource that is currently exploited in many applications and services based on semantic web technologies. A vast amount of such information has been created by memory institutions in the last decades using different standard or ad hoc schemas, and a main challenge is to make this legacy data accessible as reusable semantic data. On one hand, this is a syntactic problem that can be solved by transforming to formats that are compatible with the tools and services used for semantic aware services. On the other hand, this is a semantic problem. Simply transforming from one format to another does not automatically enable semantic interoperability and legacy data often needs to be reinterpreted as well as transformed. The conceptual model in the Functional Requirements for Bibliographic Records, initially developed as a conceptual framework for library standards and systems, is a major step towards a shared semantic model of the products of artistic and intellectual endeavor of mankind. The model is generally accepted as sufficiently generic to serve as a conceptual framework for a broad range of cultural heritage metadata. Unfortunately, the existing large body of legacy data makes a transition to this model difficult. For instance, most bibliographic data is still only available in various MARC-based formats which is hard to render into reusable and meaningful semantic data. Making legacy bibliographic data accessible as semantic data is a complex problem that includes interpreting and transforming the information. In this article, we present our work on transforming and enhancing legacy bibliographic information into a representation where the structure and semantics of the FRBR model is explicit.

Keywords: Cultural Heritage, Data Translation, Semantic Interoperability, FRBR, XML, MARC

1. Introduction

Information about cultural objects is a major point of interest on the Web and in recent years there has been a significant change in the way we want to disseminate and reuse this information. Semantic web technologies can be used to expose and interpret the meaning of the data, open access enables third parties to develop innovative new services for existing data, and new knowledge can be created by linking related and complementary data from different sources. A large portion of our cultural heritage is already thoroughly documented by memory institutions worldwide, but to realize the potential value of this metadata, there is often a need to migrate or transform this information into a representation that enables semantic aware reuse and integration [23,28].

Libraries have for decades created metadata records describing the products of intellectual and artistic endeavor expressed as text, music or other forms. The use of this metadata has traditionally been limited to library services, but it is a resource that could be

*Corresponding author. E-mail: takhirov@idi.ntnu.no.
**The author carried out this work during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme.

exploited in innovative ways beyond the library domain [14,27]. Bibliographic information systems intersect with numerous other available resources such as Wikipedia or Freebase, sites for specific artists, authors or genres, social sites devoted to discussing literature and music, as well as many of the online stores. Most libraries are public institutions, and there are few commercial and legal barriers for making library records open access. The information that is managed by libraries is an important global documentation of the intellectual and artistic endeavor of mankind that could and should be reused and integrated with other sources in innovative services that enable users to learn about, discover, annotate and discuss our cultural heritage [4].

However, a major problem of current bibliographic information is that it is difficult to exploit by others. The formats used are variants of MARC, the format developed in 1960's by Library of Congress. This format was originally designed to transfer card catalogues to magnetic tape. The concepts used for the data elements are not always meaningful in other contexts and the information is mainly intended to support known-item searching and display of records on screen or in print. The library community has generally recognized the need for modernizing the rules and standards and the conceptual ER-based model in the Functional Requirements for Bibliographic Records (FRBR), is an important foundation for this renewal [40]. The library environment is unfortunately inherently conservative and change resistant because of the huge number of existing collections, systems and practitioners involved worldwide [39]. A major hurdle for adopting new models is the amount of existing legacy data and the many challenges that are related to the reinterpretation of this information in the context of the FRBR model.

Our main research objective is to explore the interpretation of bibliographic information and migration of bibliographic databases to a new information model and the main goal is to increase the value of this information. Our approach is to make explicit the entities and relationships that can be extracted from bibliographic records which consequently will enable semantic aware integration and reuse as well as new services based on this information. In this article, we present a general framework, FRBR-ML[1], for managing and evaluating the conversion of existing MARC-based data into a representation that is based on the FRBR model. Conversions are performed using a system that can be adapted to any dialect of the MARC format. The output of the conversion consists of a distinct set of FRBR entities and relationships. Indeed, conversions will typically include many errors because we are transforming information, which was primarily created to be human readable, into a more stringent model with explicit entities and relationships that can be machine interpreted. To evaluate the conversion results, we use metrics for completeness, redundancy and extension. Another important part of our research is to exploit the use of enhancement and correction techniques to improve the result. The format in FRBR-ML builds on the MarcXchange standard for coding the attributes from the original records, and introduces additional elements for grouping fields into typed entity descriptions with support for identification and referencing combined with different solutions for expressing typed relationships. The format is compatible with RDF/OWL by direct transformation and we show that the format supports a round-trip transformation to an alternative representation in MARC that is capable of expressing the structure and semantics of the FRBR model.

The contributions of this article can be summarized as follows:

- we propose a framework that includes an exchange format with adequate properties such as readability, simplicity and understandability with respect to the FRBR model;
- we semantically enrich and enhance legacy data by using external knowledge base and services;
- we propose evaluation metrics to evaluate the quality of the transformation;
- we run experiments with a dataset of Norwegian national bibliography to show the benefits of our approach.

The article is organized according to this outline. We present the background information in Section 2. Then, in Section 3, we discuss the related work. In Section 4, we present two use cases, formalize the problem related to the conversion of MARC records and provide an overview of our framework. Next, we detail the different parts of our FRBR-ML framework: representation (Section 5), semantic enrichment and correction (Section 6), and design metrics (Section 7). To evaluate our approach, we describe in Section 8 the experiments performed with the Norwegian national bibliography. Finally, we conclude and outline the future work in Section 9.

[1] FRBR-ML stands for FRBR in XML.

2. Background

In this section we provide the background information: we introduce the characteristics of bibliographic information and continue with MARC format which is widely used as a storage format for library records. Finally, we present the FRBR model along with FRBR ontologies and vocabularies.

2.1. Bibliographic Information

The underlying principle in cataloging is that an item (e.g. a book) should be described using the information that is found on the item and the bibliographic record should include the access points that are needed by users for searching. In the description, one typically distinguishes between main and added entries for titles, persons and corporate bodies. The main entry is the primary access point the record should be organized under - often the author - and added entries are other access points that users should be able to find the record under. This framework has the origin in card catalog where one had to decide what catalog cards to create for a particular publication. Although computer catalogs are now the norm, the current cataloging rules originate from the card catalog era. The entries for persons (and corporate bodies) are authority controlled, to ensure that names are used consistently throughout the catalog.

2.2. MARC

Exchange of bibliographic data is an important service in the library domain and the MARC format is a key bibliographic standard for the flow of information between libraries [29]. MARC is an acronym for MAchine Readable Cataloging and is an encoding format that was developed in the late 60's at the Library of Congress. It is an implementation of ISO 2709 [24].

Although MARC can be considered as merely an exchange format, its impact on the final structuring of information is significant. MARC formats are used both for input and as the internal logical and physical data model in many systems. Many variants or dialects of MARC have been developed since then. Two formats are particularly important due to their international character: UNIMARC, developed by IFLA as an international exchange format, and MARC 21, the most widely used format. In the rest of this paper MARC is used for MARC 21. Each MARC record, such as the one depicted in Figure 1, typically describes a single publication and each datafield reflects a logical grouping of the data elements that together describe a specific aspect of a publication. For instance, publication information is stored using the MARC datafield 260, with three subfields for place of publication ($a), publisher ($b) and publication year ($c). Records are self-contained information units which means that each record contains all the information needed for a publication.

```
001  ocn310152465
020  $a9780007208661 (pbk.)
020  $a0007208669 (pbk.)
041   $aeng
100  1 $aChristie, Agatha,$d1890-1976.
245  10$a1960s omnibus :$bEndless night, By the pricking of my
thumbs, Passenger to Frankfurt, Postern of fate /$cAgatha Christie.
246  3 $aNineteen sixties omnibus
260   $aLondon :$bHarperCollins,$c2006.
300   $a775 p. ;$c20 cm.
650  0 $aDetective and mystery stories, English.
700  1 $aChristie, Agatha,$d1890-1976.$tEndless night.
700  1 $aChristie, Agatha,$d1890-1976.$tBy the pricking of my thumbs.
700  1 $aChristie, Agatha,$d1890-1976.$tPassenger to Frankfurt.
700  1 $aChristie, Agatha,$d1890-1976.$tPostern of fate.
```

Fig. 1. A MARC Record Describing a Book with Four Novels.

The core ISO 2709 standard defines a generic and somewhat simple structure, but a bibliographic record itself can be rather complex. The different tags of the datafields reflect the main concepts used in cataloging but their meaning can depend on type of material cataloged and may vary between records. Indicators or specific subfields are used to differentiate the meaning but are often inconsistently used [1]. In our example, the datafield "100" ("Main Entry–Personal Name") has an indicator with value "1" ("surname is represented first") and a subfield "$a" ("Personal Name") with value "Christie, Agatha". Some tags and codes are defined as not repeatable (e.g., the control field "001" normally used as a local identifier) while others can be repeated (e.g., added entries represented in "700" fields) and the sequence may be significant. Another characteristics of bibliographic records is the use of strings or codes as the only data type. All information – even dates and numerical values – are coded using a mixture of numerical values, punctuations, separator characters and abbreviations.

In the recent years, there has been an increased interest in creating more advanced end user services for exploring the contents of library catalogs and libraries are starting to realize that the information model used for current bibliographic records may have to be changed to be able to adapt to new requirements. A library record can be a very rich source of knowledge about many aspects of a publication, but most of this knowl-

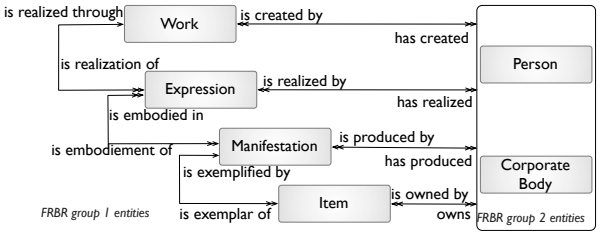

Fig. 2. FRBR Group 1 and Group 2 Entities

edge is unfortunately only released after the record is found, when it is displayed to the end user.

To make MARC records available to a wider range of stakeholders, the Library of Congress developed the MARCXML format for encoding MARC 21 records using XML [32]. This standard is often referred to as lossless as it enables a round trip conversion *MARC 21-MARCXML-MARC 21* without losing information. MarcXchange is a corresponding ISO standard [25] that is more generic and supports any ISO 2709 compliant MARC format. Metadata Object Description Schema (MODS) is another XML-based derivative of the MARC 21 format that includes a subset of MARC 21 fields with language-based elements and attributes as content designators [31].

2.3. FRBR

The ER-based model presented in the IFLA report on Functional Requirements for Bibliographic Records (FRBR) [8] is a major step towards modernization of bibliographic records and cataloging. The model identifies the main entities and relationships that are of interest to end users and it is designed to support the following four tasks: *find* entities that correspond to user's expressed information need, *identify* entities, *select* entities and *acquire* access to entities.

The FRBR model depicts intellectual products as four interrelated entities: *Item*, *Manifestation*, *Expression* and *Work* (Figure 2) [2]. *Manifestation* and *item* entities are equivalent to the commonly known concepts of publication and copy respectively. The intellectual contributions found in publications are modeled in FRBR as the *expression* and *work* entities. A *manifestation embodies* one or more expressions whereas each expression *realizes* a single work. An *expression* is the intellectual product that we recognize as unique content in the shape of text, sound, images or other types independent of the specific formatting it has been given in different *manifestations*. The *work* entity is the most abstract and is needed because of the way we refer to and reason about intellectual and artistic creations at a more general level. The play by Shakespeare commonly referred to as "Hamlet" exists in numerous translations where each translation is considered to be a specific *expressions* which *realizes* the same work. The main advantage of the *work* entity is that it enables collocation of intellectually equivalent products and enables the modeling of closely related intellectual products in tree-like structures. The FRBR model additionally includes entities for agents (*person* and *corporate body*) and the relationships they have to the different intellectual and physical products. Shakespeare *created* the *work* Hamlet, and the person responsible for a specific translation is related to a particular expression with *has realized* relationship.

On one hand, FRBR model is considerably different from the data structure that is found in MARC records. On the other hand, the different entities are often implicitly or explicitly described in the bibliographic records. As an example, the MARC datafield 245 (subfield $\$a$) in Figure 1 is the title of the publication and is normally considered an attribute of a FRBR *manifestation* entity. The FRBR model is not intended to serve directly as a data model, but there has been a significant interest in the use of the model as a foundation for new types of services and user interfaces. As a conceptual model, the main contribution of FRBR is a more knowledge-like representation of bibliographic data that enables many types of applications such as exploratory interfaces where users are presented with listings of works for each author and can follow the relationships to learn about and find the versions or editions they prefer.

The FRBR report was published over a decade ago and has so far not been extensively implemented in library systems. However, the model is far from being neglected; the promising Linked Open Data (LOD) vision and the increasing demand for semantic data has revitalized the interest in the model. Additionally,

[2] All FRBR work, expression, manifestation entities are presented in *italic* font to avoid misunderstanding.

there is a number of prototypes and production systems available that are at least partially based on the model. Finally, the FRBR model is rather simple to implement if one does not have to consider compatibility with the existing data.

2.4. FRBR Ontologies and Vocabularies

One of the first RDF vocabularies based on the model was published in 2005[3]. The need for a more official vocabulary is acknowledged by IFLA and there is ongoing work to establish a vocabulary under the IFLA namespace[4].

The FRBRoo ontology is a different approach to the formalization of FRBR as an implementation model [15]. The underlying idea behind this ontology is to merge the FRBR model with the standardized CIDOC Conceptual Reference Model (CRM) that provides definitions and a formal structure for describing the concepts and relationships used in cultural heritage documentation [30]. The merged ontology expresses FRBR with the formalism used in the CRM model and it is intended to facilitate the integration, mediation, and interchange of bibliographic and museum information, as well as elaborate and clarify the more pragmatic or unclear parts of the FRBR model.

3. Related Work

Different experiments, prototypes and systems have in the last decade attempted to interpret existing MARC-based information using the entities and relationships defined in the FRBR model. One of the earlier experiments was performed by **Hegna and Murtomaa** using the Norwegian and Finnish national bibliographies and they demonstrated how existing data elements could be used to identify the *work* and *expression* entities for selected authors [21]. An important problem identified in this study is the difficulties in interpreting records with added entries which are often used when the content consists of multiple individual parts (books with two or more novels, essay collections, etc.). Another problem they discuss is the inconsistent use of the key information that is needed for identifying *works* and *expressions* correctly.

One of the algorithms for interpreting MARC-based records using the concepts introduced in the FRBR model, is the **OCLC work-set algorithm** [22]. It is developed for records in the MARC 21 format and implements a strategy for selecting work-related information which is used for clustering records that describe the same *work*. The algorithm basically treats all records as describing a single *work*, which partly is a consequence of the MARC 21 format and current cataloging practice that seems to favor the use of descriptive contents notes rather than structured added entries when cataloging publications that have multiple distinct parts – such as books containing multiple novels and essay collections. The FictionFinder[5] prototype, which makes use of the algorithm, demonstrated how the FRBR model can be used to create listings of works by author and additionally supported browsing by genres, characters, settings and literary awards.

Later experiments include the **TelPlus project** that processed records related to Nobel laureates [33] from different catalogs including both MARC 21 and UNIMARC. The algorithm for clustering works is somewhat comparable to OCLC work-set algorithm but with a different approach to identify and merge equivalent entities based on string matching. The prototype manages different MARC formats and it treats expressions and manifestations as distinct entities. A prototype user interface demonstrating the results was evaluated in terms of usability.

The Variations project at the Indiana University Library has a different approach as they attempt to interpret records related to music using a strategy for interpreting added entries as separate entities [36]. Music is often cataloged with a more extensive use of added entries for titles and performers and composers of the various tracks on a CD. The prototype looks up the main title to identify if it is a collective title (generic uniform title) in which case the added entries are interpreted as the main content instead. The Scherzo prototype[6] can be used to search for and explore the catalog by composer, performer, instrumentation and work.

The FRBRizer approach [2] that we have developed to extract FRBR entities and relationships is based on the assertion that a proper interpretation of bibliographic records is best solved by distinguishing between the extraction of the entities and relationships in each record. To create a complete interpretation of a MARC record, we need to interpret all fields and infer what entities the record describes and how they are related. Although a majority of bibliographic records describe a simple structure that consists of a single

[3] vocab.org/frbr/core.html
[4] metadataregistry.org/schema/show/id/5.html
[5] www.oclc.org/research/activities/fictionfinder
[6] webapp1.dlib.indiana.edu/scherzo/

work *realized through* a single expression *embodied in* a manifestation – which is somewhat straightforward to extract – there are many records having complex structure to introduce significant noise in the result if they are misinterpreted. Additionally, a simple interpretation will often ignore many of the works that are of main interest to end users. The FRBRizer tool we are using to extract FRBR entities and relationships from bibliographic records was initially developed for an experimental conversion of the Norwegian BIBSYS database [1] and has later been further developed to support more advanced interpretations. The solution is generic in the sense that rules can be adapted to any MARC formats and we are able to interpret all possible occurrences of entities including works and persons that appear in subject entries, added entries, contents notes and series statements etc. This tool uses MARC records in XML as input and produces a set of FRBR-based records with relationships expressed as links.

Different solutions for interpreting MARC as FRBR are important contributions because they can be used to migrate MARC data to other formats or can be used to correct or enhance MARC records and in this way enable the future implementation of the full FRBR model in current catalogs [9]. An important aspect of interpreting or converting MARC as FRBR that to our knowledge has not been addressed, is the assessment of the result. Projects and experiments have explored the possibilities for interpreting records and have looked at different solutions for clustering or merging equivalent entities, but to compare and evaluate different strategies and approaches, we need systematic ways to determine the level of quality that can be achieved by different strategies and techniques.

4. Overview of the Framework

We first illustrate the motivation behind our work by describing some use cases. The next part deals with the formalization of the conversion of MARC-based data into a FRBR compatible form. Finally, we introduce a global overview of our framework FRBR-ML.

4.1. Use Cases

In this section, we discuss two different types of organization of information in bibliographic records. As illustrated in Figure 3, the simplest example which accounts for most of the records is when a *manifestation* is embodying a single *expression* of a *work* [6]. In this figure, we see that *a 2006 book* (manifestation), entitled *The Road* (work) has been published in English (expression) by American writer *Cormac McCarthy* (person). The single entities from FRBR groups 1 and 2 are present, thus making the process of recognition and identification easy.

Other records are usually popular *works* that have either been repeatedly published, translated or aggregated in various records (e.g. as collection of stories) and/or in different languages. Therefore, these publications are of particular interest and they mostly benefit from the FRBR support.

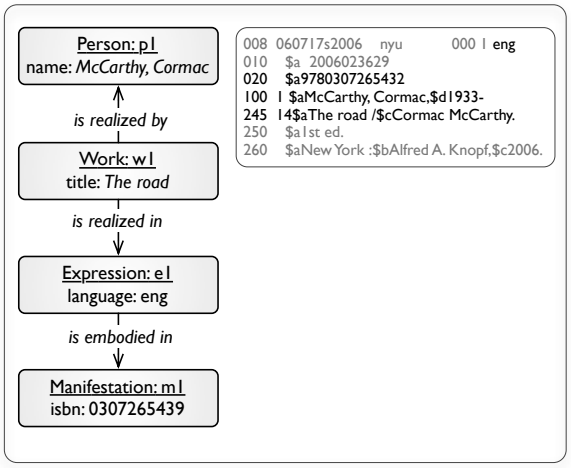

Fig. 3. A simplest case: a manifestation that embodies a single expression of a work that is created by a single person. The corresponding MARC record is shown on the right.

Multiple intellectual contributions contained in these publications are characterized as distinct intellectual entities [10]. However, not all of these entities are considered equally important: *works* such as cover art, text on the back of the cover etc. are of little interest to most end users. The *expressions* that are of concern to the most users of bibliographic information are those that we consider to be the main content such as the novels in a book containing multiple novels, the essays in an essay collection, the tracks of a music compact disc. Figure 4 describes a such complex case. In this example, German translations of two Swedish novels, we show what kind of entities and relationships are found in existing bibliographic catalogs. These *works*, “Den vedervärdige mannen från Säffle” and “Det sultna rummet”, are both originally created collaboratively by two Swedish authors “Per Wahlöö” and “Maj Shöwall”, but only the latter is mentioned in the

record as the creator (field 100 $a) of the *work*. This may be a rather unusual and challenging case, but it includes a kind of structure that we believe need to be supported in FRBR.

```
100 1$a Sjöwall, Maj, $d 1935-
24014 $a Den vedervärdige mannen från Säffle.$l Tyska
245 14    $a Das Ekel aus Säffle; $b Verschlossen und
verriegelt : zwei Romane / $c Maj Sjöwall, Per Wahlöö
700 12 $a Wahlöö, Per, $d 1926-1975. $t Den vedervärdige
mannen från Säffle. $l Tyska
700 12 $a Wahlöö, Per, $d 1926-1975.
700 1$a Schultz, Eckehard
700 1$a Maass, Hans-Joachim
740 4$a Det slutna rummet
```

Fig. 4. An ambiguous MARC record which describes a manifestation that embodies multiple expressions of different works.

As we can see from the above example, added entries (700 fields) are used as additional access points to the bibliographic record to improve the searching for records as well as to present more extensive information about the item described. Such information includes persons and corporate bodies in addition to titles that are relevant as access points. Added entries are recorded using field tags 700, 710, 711, 730 and 740 in MARC 21. With the exception of 740, these titles and names of persons and corporate bodies should – according to the rules – be under authority control. In the context of FRBR, these added entries may reflect quite different aspects of the model, e.g. in this case there are two authors of a novel. Other common usages of added entries are to include additional persons such as translators and illustrators. The type of relationship between a person or corporate body and a resource is indicated by the use of relator codes or terms [34], but the actual use of relator codes greatly varies depending on the local cataloging policy and practice. The titles found as added entries may identify *work* and *expression* entities that are related to the cataloged item in different ways such as the novel upon which a movie is based. In other cases, added entries are used for analytical entries which can be interpreted as information about the embodiment of additional *expressions* in the *manifestation*.

4.2. Formal Model

In this section, we formalize the problem of converting MARC records into an XML format. We have a **collection of records** $\mathcal{R}$ regardless of representation form. A record $r \in \mathcal{R}$ is composed of a **set of properties** $\mathcal{P}$, i.e.:

$$\forall r \in \mathcal{R},\ r = <\mathcal{P}>$$

Each property $p \in \mathcal{P}$ is represented by a name and a value. An example of a property is a MARC datafield (245 $a, The Fellowship of the Ring) where 245 is a datafield tag, $\$a$ is a subfield code, together forming a property for the *main title* of a bibliographic record. A subset of $\mathcal{P}$ provides a description of an entity. For instance, a *manifestation* entity found in a record describing "The Fellowship of the Ring" book by Tolkien, may be described by the properties title (245 $a, The Fellowship of the Ring) and manifestation identifier[7] (020 $a,0618574948) . A specific property id uniquely identifies the record r.

A record describes one or more publications and therefore can be seen as an abstraction of a **set of entities** $\mathcal{E}$, such as such as a *manifestation*, *expression*, *work* and related *persons*. Although entities are present in MARC records, they can only be extracted based on our interpretation of their properties and relationships. In contrast to that, the entities in FRBR are clearly defined by the model. Finally, these entities are related to each other through a **set of relationships** $\mathcal{L}$, such that:

$$\forall l \in \mathcal{L}, \quad e_1, e_2 \in \mathcal{E}, \quad l : e_1 \times e_2$$

4.3. Workflow of FRBR-ML

Although memory institutions are interested in semantic formats, the transition cannot be automatically achieved due to the complexity of adding well-defined semantics to a large body of existing legacy data. Indeed, ontology languages such as OWL/RDF have a great degree of machine readability which enables automatic reasoning, but converting legacy data using the correct semantics of these languages is still an unsolved challenge [12,38]. Consequently, there is a need for an intermediary format to ensure a seamless transition that enables both legacy data and semantic formats to coexist. Needless to say, the format should enable the exchange of records between different applications or services. Another challenge of the MARC format is that the specific meaning of a datafield often is contextual and not explicitly defined, thus making it difficult to automatically process information. For example, if there is a 740 *Added Entry* field we know that this is a title, but unless the second indicator has the

[7] In this case an ISBN number of the published book

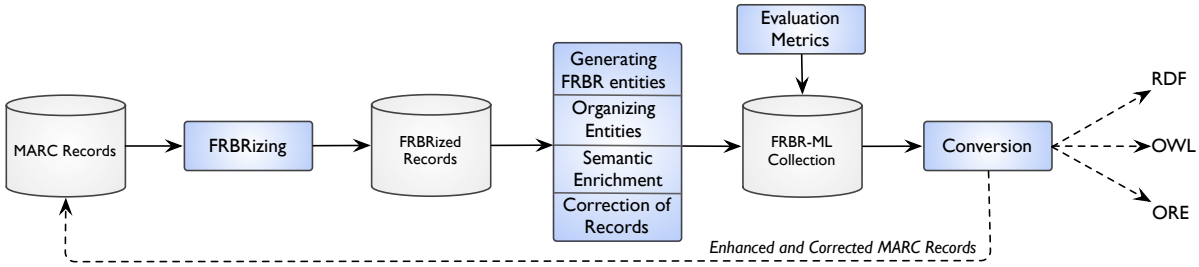

Fig. 5. Workflow of FRBR-ML.

value 2, we do not know in what way this title is related to the work(s) descibed in the record. Our FRBR-ML framework addresses the issues of interoperability and lack of semantics. Figure 5 depicts the workflow of the process of transforming MARC records into FRBR-ML format. The process starts with the conversion of MARC records using an extended version of the FRBRizer tool [1,3]. This tool performs the conversion of MARC records into a set of FRBRized records using a set of pre-defined rules and a series of XSLT transformations. An example of the output of the conversion is shown in Figure 6.

```
<record label="Work" type="C001" id="76d632d21tr0w1bwf2gf4e1ed4f">
  ...
  <datafield tag="245" ind1="1" ind2="0">
    <subfield code="a">Tolkien</subfield>
  </datafield>
  <relationship type="P2001" label="is realized through"
    href="57d655d21bd478bdf264e1ed737ac02c"/>
  <relationship type="P2009" label="is created by"
    href="e4a694dc75d1dff64fee6e90ad61a06b"/>
  <relationship type="P2033" label="has as subject (person)"
    href="be23140269fe12e16b8e8b8007a8192c"/>
  <keyvalue>whitemichael#tolkien#</keyvalue>
</record>
```

Fig. 6. A fragment of the output of the FRBRizer tool.

The next step is to convert the FRBRized records into our **intermediary and explicit format**. Contrary to the output of the FRBRizer tool, our format aims at reflecting the full structure and semantics conveyed by the FRBR model. Similarly to [35], we strive to maintain the human readability while allowing an easy transformation into a machine interpretable form. In addition, the FRBR-ML format uses the FRBR vocabulary to promote simplicity and understandability, since libraries and memory institutions are increasingly interested in adopting FRBR in the long term. Once our tool has converted the records into structured FRBR entities, it **enriches** them by querying external resources and it performs **correction** of properties. The result, stored in the FRBR-ML format, can further be converted either to RDF, OWL, ORE or back to enhanced MARC records. Our intention is to support a **two-way interoperability** between systems that manage resources in a variety of formats. Finally, to identify interesting properties of our format, we have defined **three design metrics** to measure the loss of information, the amount of redundancies and the percentage of semantic enrichment.

In the next sections, we present in more detail the different parts of our framework: representation of the format and interoperability (Section 5), semantic enrichment (Section 6), and design metrics (Section 7).

5. FRBR-ML: Representation

FRBR-ML features an entity-oriented representation that is comparable to new knowledge representation frameworks based on resource descriptions. However, for managing this information we will argue that there is a need for a simple and understandable format. The end-user of FRBR-ML would be people who manage cultural heritage and therefore used to managing record-based information. Our approach allows us to bridge the gap between record and resource based representations. Furthermore, using an XML based representation would enable a lower barrier for understandability (Section 5.1).

The second feature of FRBR-ML is that it represents MARC-based information with a clear structure. By clear structure, we mean that the information is encoded following the FRBR specifications. This **organized representation** should also minimize the loss of data and redundancies (Section 5.2).

The last point is related to **exchange of records**. The representation should ensure compatibility with both record-oriented and semantic formats (Section 5.3).

5.1. Entity-oriented Representation

To facilitate understandability, we are adapting the entity-oriented representation to the record-oriented, embedding the FRBR entities in the records with their respective FRBR semantic FRBR type. Instead of representing FRBR entities with records, they are embedded within their respective semantic FRBR type. The naming convention for semantic types in our representation framework follows the FRBR model, making navigation and browsing simpler.

The FRBRized collection is represented as a set of XML documents. In these documents, each root element labeled "collection" has *work*, *expression*, *manifestation*, *person* and *corporate bodies* as child elements. Each of these elements contains properties represented by datafields and control fields embedded under a semantic element μ. This semantic element is provided by a map function that takes as an input a property p:

$$\mu := map(p)$$

If the function returns the same semantic element for several properties, all of them are grouped under this semantic element. This function will be discussed in more details in Section 6.1.

In FRBR-ML, entities are linked by relationships. These relationships are not transitive, as defined in the FRBR model. However, all of them are bidirectional. For example, a person who created a work does not mean that (s)he created a manifestation as well. But a person has a relationship *isCreatorOf* to work and the work an inverse relationship *isCreatedBy* to that person.

Figure 7 depicts a fragment of a *manifestation* entity. Like any element representing a FRBR entity, the manifestation element can contain a set of attributes, a set of MARC fields and a set of semantic elements. These semantic elements include values of properties of a FRBR manifestation entity as well as semantically enriched information (see Section 6.1). A similar representation pattern is used to describe *work*, *expression* and *person* entities, as shown in the complete schema[8].

Since the same entity may occur in multiple records, there is a need for discovering equivalent entities. This is mainly intended to solve the redundancy problem in the output by merging entities that are very likely to be the same. As pointed out in the works of others [33], more flexible techniques are needed to determine if two entities represent the same real-world entity and should be merged into one. However, we believe that it is important to have a complete and correct interpretation of the entities and relationships described in the record as a first stage, then followed by matching techniques in combination with verification and correction techniques in order to improve the quality of the results. Our approach is based on comparison of entities by the use of key descriptive information that is unique for each entity within the collection based on selected property values. We generate identifiers for these entities by calculating MD5 hash for their corresponding their corresponding key values. A post-processing step merges identical entities based on this hash value, which eliminates duplicate entities.

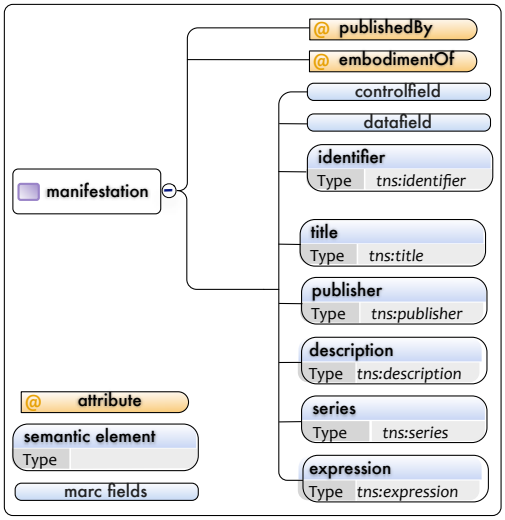

Fig. 7. A fragment of schema depicting a manifestation entity.

Having finalized the element level discussion, we detail how we organize entities.

5.2. Structural Organization

Expressing relationships in a well-defined manner between entities in XML is one of the important tasks to avoid duplication and loss of data. Furthermore, it is a first step to introduce semantics in existing data. In the rest of this section, we study the different possibilities to represent relationships between entities, namely **hierarchical** and **referencing**. Finally, we describe our solution, a **hybrid** method that combines advantages of the aforementioned ones.

[8] www.idi.ntnu.no/~takhirov/frbrml_hybrid.xsd

5.2.1. Hierarchical Method

As the name implies, this method enables the expression of entities and their respective relationships with hierarchical organization (also called parent-child relationship) [18]. One of the first advantages is an

```
<manifestation id="m1">
  <expression id="e1">
    <work id="w1">
      <person id="p1">
        ...
      </person>
      ...
    </work>
    ...
  </expression>
  ...
</manifestation>
```

Fig. 8. An Example of Hierarchical Method.

increased readability with implicit semantics between FRBR entities. For instance, a *manifestation* that includes a child element *expression* has an implicit "isEmbodimentOf" relationship to that expression (the inverse of *isEmbodiedIn*). Other advantages are compactness and proximity of data, which enable faster processing. On the other hand, this method suffers from possible infinite loop (e.g., an entity having a relationship to an ancestor entity in the tree). It is also insufficient to represent more than one relationship type under the same parent. Another disadvantage of this approach is data duplication. This issue is present when there are several relationship references to the entity. Figure 8 depicts an example of this hierarchical method in which all FRBR entities are represented as nested elements with the manifestation as the root. Having the manifestation as the topmost node in a hierarchical representation may seem to contradict the FRBR model that has *work* as the the most abstract item, but the model does not describe any particular arrangement of the entities. Using the manifestation as root element corresponds to to the traditional way of organizing metadata into a record for each of the described items. This arrangement is also tailored the creation of metadata which typically is a process of describing the entities and relationships that makes up the description of a specific manifestation. It is also the arrangement that would lead to the most evenly distribution of entities in records as the maximum number of expressions a manifestation embodies is rather low compared to the inverse case. Other hierarchical arrangements would be better suited for other situations such as using *work* as the topmost level when presenting search results to users who want to explore what is available for a specific author, but such representations can easily be created on the fly when processing a collection of records.

5.2.2. Reference Method

We can also employ the reference method when expressing relationships between entities [26]. This method is based on the usage of ID/IDREF similarly to URI in RDF. There are different techniques for referencing entities. A first technique is *dynamic typing*, i.e., the type of the relationship is specified by an attribute *type* of a given *relationship* element. It provides a greater flexibility when one needs to add new relationships. The second technique is to have a set of *strongly* or *statically* typed predefined relationship types which are represented as elements. Although this technique eliminates the readability weakness of the first technique, the problems arise when there is a need to define new types of relationship.

```
<manifestation id="m1" isEmbodimentOf="e1">
  ...
</manifestation>
<expression id="e1" isRealizationOf="w1">
  ....
</expression>
<work id="w1">
  ....
</work>
```

Fig. 9. An Example of Referencing Method.

No matter which technique is used, the reference method avoids duplication and it provides a better support for updating data. Furthermore, it reduces the size of the XML document. However, related entities are stored in a loosely coupled manner which means it is less efficient in terms of processing. Indeed, entities are spread across the document and accessing them requires more effort. As illustrated in Figure 9, all entities are stored under the root element with this reference method. We notice that the manifestation is linked to an expression entity with an *isEmbodimentOf* reference.

5.2.3. Hybrid Method

The main drawbacks of the hierarchical approach are data duplication and the constraint related to expressing single relationship type between entities. On the other hand, the reference method is less efficient in terms of processing. Our hybrid approach combines the representation of both methods. In one case, an en-

tity can be stored hierarchically under its related entity taking advantage of proximity, readability and efficient processing. But under different conditions, an entity is stored using the reference method to avoid duplication if it appears several times in the collection. For example, a particular embodiment of expression can be represented as either child element (hierarchical) or by attribute (referencing method).

The relationships between entities are represented using the strongly typed method (see Section 5.2.2). This method has been chosen due to the fact that a set of predefined relationship types are specified in the FRBR model.

The process of deciding which representation method to use for a given entity is shown in Algorithm 1. This process takes as an input a set of records $\mathcal{R}$. We initialize a set of constraints λ that should be respected for hierarchical representation (line 3). A constraint is a tuple containing two entity types and a relationship type, indicating that this relationship is allowed. All records are analyzed and for each of them we extract their entities (line 5). For each entity, the decision to represent it hierarchically or referentially is made by the $decide$ function (line 7) before adding it to the set of entities $\mathcal{E}$.

In the $decide$ function, we obtain the set of relation types for the given entity e (line 13). To avoid infinite loop, we add this entity to the current stack of entities (line 14). ΔT contains all distinct relation types t of e. The next step analyzes each of these relation types to discover the set of entities C linked to e with relation type t (line 16). The decision to represent an entity hierarchically depends on whether the related entities violate any constraints (line 18) as well as the presence of the entity in the current stack. If these conditions are not met, the entity is represented using the reference method (line 23). The $decide$ function recursively iterates over entities (lines 21 and 25). The set of extracted entities with typed relationships is then stored as a FRBR-ML collection in XML.

The hybrid algorithm is flexible due to the set of constraints. By default, our set of constraints includes the basic relationships defined in the FRBR model. However, one may define additional constraints to meet specific requirements.

Algorithm 1 Hybrid representation decision.

Require: Set of Records $\mathcal{R}$
Output: Set of Entities $\mathcal{E}$
1: **function** start()
2: $\mathcal{E} \leftarrow \emptyset$
3: $\lambda \leftarrow constraint\_definitions()$
4: **for all** $r \in \mathcal{R}$ **do**
5: $E = extract\_entities(r)$
6: **for all** $e \in E$ **do**
7: $decide(e)$
8: $\mathcal{E} \leftarrow e$
9: **end for**
10: **end for**
11: **end function**
12: **function** decide(e)
13: $T = get\_relation\_types(e)$
14: $add\_to\_stack(e)$
15: **for all** $t \in \Delta T$ **do**
16: $C = find\_conn(e, t)$
17: **for all** $c \in C$ **do**
18: $\delta = violates\_constraint(e, c, t, \lambda)$
19: **if not** δ **and not** $in\_stack(c)$ **then**
20: $hierarchical\_rep(c)$
21: $decide(c)$
22: **else**
23: $referencing(c)$
24: **if not** $in\_stack(c)$ **then**
25: $decide(c)$
26: **end if**
27: **end if**
28: **end for**
29: **end for**
30: **end function**

5.3. Exchange of Records

The output of the final transformation is a set of entities described with clear structure as well as typed relationships. These entities are assigned the same identifiers as those generated by the FRBRizer tool. They have relationships to other entities in the same collection by using either referencing or hierarchical method. This output can be converted to other representation formats such as RDF/OWL as well as more domain specific formats such as MARC. These conversions are performed by a series of XSLT transformations.

We begin with describing the **transformation to RDF and OWL**. The RDF conversion is represented using the vocabulary provided in [13]. This vocabulary is an expression of the concepts and relations described in the FRBR model in RDF. Properties follow naming convention such as *Et dukkehjem has_realization A doll's house*. In this vocabulary, most properties are paired with an inverse, e.g. *embodiment/isEmbodimentOf*. The use of synthetic superclasses for ranges and domains are discarded and we only employ concrete entity types. In other words, we relate *works* directly to *person/corporate body* rather than to ResponsibleEntities. Furthermore, this vocabulary uses OWL-DL to provide constraints. As the vo-

cabulary lacks support for attributes of the entities, we additionally use the FRBRer model vocabulary [16] to describe attributes. As for OWL, we generate an OWL instance via XSLT transformation for each XML document validated by the schema. The FRBR entities *work, expression, manifestation, person* etc. are declared as *owl:Class* and *owl:ObjectProperties* to specify elements and attributes of the entities. While we specify the bidirectional relationships for RDF, we can make use of *owl:inverseOf* construct for OWL transformation.

An extension that we have implemented in the FRBRizer tool deals with the storage of the MARC control field 001 in order to ensure a correct **transformation back to MARC**. The control field 001 contains the unique control number assigned by the organization creating, using, or distributing the record. All datafields and control fields have an element $<mid>$ specifying the original control field 001. During the transformation back to MARC, we can use this information to correctly construct the original MARC record since some records might have been merged. The process of transformation back to MARC starts at the manifestation level. We collect all datafields and control fields that are pertinent to the record, i.e., those with the same $<mid>$ value. That is, we traverse the conceptual tree of entities from *manifestation* to *person/corporate body*, *expression*, and *work*.

As for the **transformation to ORE**, we are concerned with the description of aggregations of entities. For each record related entities identified by the previously mentioned control field 001, we can collect all the related entities which forms an aggregation. For this representation, we use the same RDF vocabulary.

As a summary, our representation ensures compatibility with record-oriented formats, such as MARC 21 but also with strongly semantic formats such as RDF and OWL.

6. FRBR-ML: Semantics

On the semantic aspect, the framework provides a balanced degree of **semantics** (Section 6.1). Furthermore, FRBR-ML includes a **correction process** to improve and disambiguate the information found in original records (Section 6.2).

6.1. Semantic Enrichment

The task of semantic enrichment is crucial in our process. On one hand, all information in a MARC record may not have a clear semantic. For instance, the datafields tagged with values 700-740 are added entries that often can be difficult to interpret the precise meaning of and associate to the correct entity. On the other hand, new formats such as RDF or OWL include a well-defined semantic to enable reasoning or complex querying. Consequently, a conversion from MARC to RDF needs to involve the identification of the precise meaning all information in MARC records. This is a complex problem and in our FRBR-ML framework we aim to solve problems related to the semantics of attributes, entities and relationships. For the attributes we use a map function to look up the correct label of datafields (e.g., *title* label for the datafield 245). More formally, the map function uses either a dictionary-based technique or a knowledge-based technique to discover the semantic element of a given property p with name p_n and value p_v:

$$map(p) = \begin{cases} dict(p_n) & \iff \exists\{dict(p_n)\} \\ knowl(p_v) & otherwise \end{cases}$$

For entities and relationships that the interpretation process is unable to interpret the exact meaning of, we primarily use the knowledge based technique to discover the correct type.

6.1.1. Dictionary-based Matching

Based on the MARC specification for the format we are processing, we build a dictionary of corresponding elements between the datafield tag, the subfield value and the semantic element. A fragment of this table is shown in Table 1. Discovering the semantic element requires a lookup in the table by decomposing a property name into a datafield tag and a subfield code. If this index pair is not found in the table or if there are multiple entries (which would indicate that there is more than one possible semantics label), it means that obtaining the semantic element for that particular index pair depends on the value of the index pair. Thus, we need a refined technique based on external resources to discover the semantic element for this pair.

6.1.2. Knowledge-based Matching

The lack of semantics is preponderant with many entries found in MARC records. Persons identified have different roles and should be related to works, expressions and manifestations in different ways. If relator codes are missing, we have to identify the appropriate type and target and endpoints of the relationship.

Property Name		Semantic
Data Field Tag	Subfield Code	Element
245	-	Title Info
245	a	Title
100	a	Name (Personal)
...	...	...

Table 1
A Fragment of our Dictionary

Titles in added entries may identify distinct *works* or be related to the *expression* or *manifestation* entities and we need to find out what type of entity they relate to. Some fields may have unidentified or ambiguous semantics and we need to interpret the exact meaning of the value. The problem of discovering the correct type of an entity, relationship or attribute value is very complex. However, we advocate that it is possible to discover the semantic type represented by this entry value, e.g., a writer or a location. To fulfill this goal, we rely on external resources for, mainly semantic knowledge bases such as *DBpedia*, *Freebase* or *OpenCyc*.

Our task is very similar to entity ranking, which consists of discovering a Linked Open Data (LOD) entity's main page. The LOD cloud refers to interconnected knowledge bases, which can be seen as the foundation of the LOD vision [20]. Many *works* have been dedicated entity ranking, such as [41,37] to name a few. In addition, two yearly challenges have an entity ranking track: *Initiative for the Evaluation of XML Retrieval* [19] and *Text Retrieval Conference* [42]. In our context, we do not have as much information as in entity ranking. Thus, we propose to discover the correct LOD entity by using **aliases**, i.e., alternatives forms of an entity's label. *J._R._R._Tolkien* is the label of the DBpedia entity representing the famous writer of Lord of The Rings, and a few of its aliases are *John_Ronald_Reuel_Tolkien*, *J.R.R_Tolkien* and *Tolkien,_J._R._R.*. These aliases are properties of an entity. For instance, Freebase provides alias for an entity (property *fb:common.topic.alias*) while DBpedia includes redirections (*dbpedia-owl:wikiPageRedirects*). Once the correct entity is discovered, it is possible to obtain its type and use it as semantic element.

More formally, we first normalize the property value p_v, i.e., replacing spaces with underscores, removing extra information in brackets, etc. As a result, we obtain a set of normalized queries $\mathcal{Q}$ for the value p_v. Given a set of knowledge bases $\mathcal{K}$, we send a query $q \in \mathcal{Q}$ against each knowledge base $k \in \mathcal{K}$ to obtain a set of ranked LOD entities. We note t_{qk} the set of result entities returned by the knowledge base k for the query q. The number of results in each set t_{qk} depends on the techniques used to query the knowledge base. Many semantic knowledge bases include various possibilities for retrieving an entity [7]:

- direct access by generating the URI of the entity. In this case, each set t_{qk} contains 0 or 1 entity;
- querying SPARQL endpoints. With this technique, the number of returned entities varies from 0 to the size of the knowledge base;
- querying a search engine or an API, which returns a set t_{qk} with any number of entities as well.

Since the knowledge bases on LOD are interrelated (property *owl:sameAs*), the same entity may appear in different result sets. We first detect which entities are identical thanks to this OWL property. We define φ as the size of the largest result set t_{qk}. To discover the correct entity, the idea is to apply statistics against all result sets. We assume that the rankings of the knowledge bases are somehow coherent and that the correct entity should appear in most rankings at the top. We therefore compute a score for a LOD entity x as follows:

$$\forall t_{qk}, score_x = \sum \min(rank(x, t_{qk}), \varphi)$$

In other words, we sum the different ranks of the entity x in each result set. If the entity does not appear in t_{qk}, the size of the largest result set with value φ is added. Finally, the entity with the smallest score is selected and its type is used as semantic element in our format.

6.2. Correction Process

The full potential of MARC records is often disregarded, which resulted in records with ambiguous semantics. Enriching with semantic information is not sufficient since it depends on identifiable entities. This is usually the case when a record does not contain enough information about entities. As we saw in the use case in Section 4.1, the translator "Hans-Joachim Maass" is not linked to any expressions that he has contributed to. Consequently, we have two tasks to accomplish: (i) identifying the type of entity to which "Hans-Joachim Maass" is linked and (ii) finding the correct relationship to the entity in the record. To achieve this goal, there are different strategies that can be employed:

- **intra collection search**. The publication may appear several times in different records of the same collection, especially for collection of *works* bun-

dled as a single *manifestation*. In that case, we can analyze such related records to find the correct relationship.

– **inter collection search**. We use external services to perform a search for the entity. This is achieved querying z39.50[9] or SRU/SRW[10] endpoints. With the use of a library catalog supporting one of the above protocols, we can find the lacking information.
– searching the **LOD cloud**. To discover the correct entity in LOD, we can use the same method as in Section 6.1.2. Then we can analyze each pair of property/values of this entity to detect an eventual relationship of the unknown entity.

If the relationship is discovered by applying any of the above methods, then we identify and enrich the entity with the relationship. When a conversion back to MARC is performed, this missing information about the entity can be corrected with regards to the initial MARC record. Indeed, an intra-collection search is more efficient to identify and find the correct relationship to the record since no network connection overhead is involved. Additionally, there is a fair chance of a match in the same collection for entities we are looking for. Searching the LOD cloud could provide good results too in terms of efficiency given the knowledge base is queried locally on the same machine[11].

Recall our example of missing information about translator "Hans-Joachim Maass". This record does not contain relator code that identifies the type of entity to which he is linked. We can find this information from the results of the various applied techniques described above. Additionally, we need to find the relationship to an entity. To accomplish this, we search for each identified entity, i.e.,work "Det slutna rummet", and exclude the other entities from the query to reduce the room for misinterpretation. Once we find the record where the two entities appear, then we can assert the relationship based on the information found in this record. Finally, for each identified entity we build a cache in order to make the process of subsequent identification faster.

7. FRBR-ML: Design Metrics

To evaluate the semantic enrichment and the representation of our format, we have defined different metrics with respect to completeness (no loss of information), minimality (no redundancies) and extension (enriched information). Applying these metrics against our format enables us to demonstrate the weak points and good properties of our approach.

7.1. Completeness

The completeness measure aims at detecting the amount of information that can be lost during transformation [5]. To be complete, the transformed records should contain all properties found in the original records. However, a few properties are more important because they are used to identify entities. Thus, we have defined two completeness measures: a quantitative completeness $quant\_comp$ that measures the amount of present properties after transformation; and a qualitative completeness that measures the amount of present entities after transformation. Both metrics are applied between an original collection $\mathcal{R}$ and a transformed one denoted $\mathcal{R}'$.

The quantitative completeness $quant\_comp$ shown in Formula 1 checks all the properties of identical records (i.e., based on their identifiers r_{id} and r'_{id}) using the hash value of the property:

$$\forall r \in \mathcal{R} \text{ and } \forall r' \in \mathcal{R}' \text{ such that } r_{id} = r'_{id},$$
$$quant\_comp(\mathcal{R}, \mathcal{R}') = \frac{\sum \frac{|\mathcal{P}_r \cap \mathcal{P}_{r'}|}{|\mathcal{P}_r|}}{|\mathcal{R}|} \quad (1)$$

We define qualitative completeness as an indication of the degree the conversion process is able to interpret all possible entities. We take into account the key properties that identify an entity. For instance, creators are identified by the fields 100 (personal name main entry) and 700 (personal name added entry). Therefore, we define the qualitative completeness formula between the two collections $\mathcal{R}$ and $\mathcal{R}'$ as follows:

$$qual\_comp(\mathcal{R}, \mathcal{R}') = \frac{\sum \frac{|\mathcal{E}_r \cap \mathcal{E}_{r'}|}{|\mathcal{E}_r|}}{|\mathcal{R}|} \quad (2)$$

However, this metric does not say anything about correctness of the results. Correctness can be evaluated by manual inspection for small collections, but for larger collections we have to use more automatic verification techniques. We discuss this verification aspect in the experiments section.

Both completeness metrics are in the range $[0, 1]$, with a 1 value meaning that the transformed records are totally complete in terms of properties and entities.

[9] www.loc.gov/z3950/ (Feb. 2011)
[10] www.loc.gov/standards/sru/ (Feb. 2011)
[11] DBpedia or Freebase dumps are freely available for download.

7.2. Redundancy

The minimality metric checks the non-existence of redundant information. In [11], minimality is defined as the percentage of extra information in a generated integrated schema. This definition does not hold in our context, since our FRBR-ML approach includes a process for enriching the original records with semantics. Consequently, we propose to measure the amount of redundant information with a first metric called redundancy. This redundant information is mainly due to the aggregation of data from similar records, e.g., with rules in the FRBRizer or during the correction process.

To detect a redundant property in a transformed collection of records, we define a set $\Delta\mathcal{P}' \subseteq \mathcal{P}'$ which contains all unique properties (according to their name and value). The following constraint is therefore respected for $\Delta\mathcal{P}'$:

$$\forall p_1 \in \mathcal{P}',\ p_1 \in \Delta\mathcal{P}' \iff \nexists\, p_2 \in \Delta\mathcal{P}' : \{p_1 = p_2\}$$

The individual redundancy of a record is the ratio between the size of the sets $\Delta\mathcal{P}'$ and $\mathcal{P}'$. The properties are compared using their hash values. To measure the redundancy of a transformed collection $\mathcal{R}'$, we sum the individual redundancies of each record and we normalize the results by the total number of records:

$$redundancy(\mathcal{R}') = \frac{\sum \frac{|\Delta\mathcal{P}'_{r'}|}{|\mathcal{P}'_{r'}|}}{|\mathcal{P}'|}$$

The redundancy metric is in the range $[0, 1]$, with a 0 value meaning that the new set of records does not contain any duplicate information compared to the original ones.

7.3. Extension

In database quality or model engineering domains, extra information may not be seen as positive. However, it enables the disambiguation and enrichment of the original data in our context. Thus, our last measure, called **quantitative extension**, computes the percentage of extra information added as a result of our enrichment process. To compute this number, we need the same Δ function which contains all unique elements of a set. Indeed, the metric would be biased if it uses redundant information. The amount of enriched information between an original record r and its transformation r' equals $|\Delta\mathcal{P}_{r'}| - |\Delta\mathcal{P}_r \cap \Delta\mathcal{P}_{r'}|$. We generalize this formula between two collections by comparing identical records:

$$quant\_extension(\mathcal{R}, \mathcal{R}') = \frac{\sum \frac{|\Delta\mathcal{P}_{r'}| - |\Delta\mathcal{P}_r \cap \Delta\mathcal{P}_{r'}|}{|\Delta\mathcal{P}_r|}}{|\mathcal{R}|}$$

The extension metric is in the range $[0, +\infty[$, with a 0 value meaning that the new records have not been enriched at all. Note that we cannot automatically assess the quality of the enrichment process due to the following reasons. For one, the human judgement is required to validate given that the ground truth is lacking. Furthermore, the enrichment process consists of two steps (the addition of a semantic type and the verification/correction of relations) which differently influence the quality of extension. However, we perform a manual evaluation of the quality of enrichment described in the next section.

8. Experimental Evaluation

This section deals with the evaluation of our approach. Experiments have been performed on a computer equipped with Intel(R) Core(TM) i7 @ 2.93GHz and 12 GB of RAM running openSUSE 11.2. We begin with the description of the NORBOK dataset used for evaluation purposes. As FRBR-ML includes a new format, we propose to check how it fulfills important design criteria such as *completeness*, *redundancy* and *extension*. The next part is dedicated to semantics, i.e. we measure the error rate caused by the enrichment and correction processes. Finally, we detail a complex use case which is commonly found in the library collections.

8.1. Dataset and Evaluation Protocol

We have performed experiments on a dataset provided by the Norwegian National Library. More specifically, this dataset is a national bibliography containing **449.063** records grouped into these types of materials:

- books, pamphlets, monographs in series, maps, computerized documents (including e-books), regardless of language;
- audio books in various languages (published in Norway from 1992);
- foreign translations of works by Norwegians (from 1978);
- foreign works about Norway and Norwegian conditions;
- complete coverage from the 1921 publication of Norwegian releases in 1978 for overseas - but a large number of older works are included.

The whole collection is stored in the NORMARC format, a dialect of MARC used in Norway. We have run the enhanced version of the FRBRizer tool to identify the FRBR entities in the records. This means that we have created new conversion rules in FRBRizer to take into account all fields in the initial records. Next, we transform the data into our format and enrich it. During the transformation process described in Sections 5.1 and 5.2, we have used a set of constraints that includes the basic FRBR relationships. The enrichment process relies on two semantic knowledge bases, *DBpedia* and *Freebase*. These bases have been queried using the three mentioned techniques in Section 6.1.2, i.e., a direct access by building an URI, queries over a search engine[12] or API[13] and with the SPARQL language[14]. Note that our approach is not limited to these bases and that we could have used other sources such as *OpenCyc*. However, DBpedia can be seen as the center of the LOD cloud by containing the largest number of connections to other data sources, and it is strongly connected with Freebase[15]. In the correction process, we have sequentially applied the three proposed techniques of Section 6.2. When a search in the local database does not provide any results, we perform an inter-collection search by using the *z39.50* protocol. If we are still unsuccessful, the correction tries to discover the missing information on the Linked Open Data cloud, namely the search services provided by DBpedia and Freebase. Based on this experiment protocol, we now detail the interesting results of our approach.

8.2. Quantitative Evaluation

In this first experiment, the goal is to detect the good points and weaknesses of our format in terms of design. Thus, we have converted the collection stored in the FRBR-ML format back to MARC. We are able to compare the resulting MARC records to the original ones with different quantitative criteria.

8.2.1. Merging Results

First, we analyze the results of the merging process detailed in Section 5.1. Table 2 provides a summary of the results for each entity type. We notice that our merging process enables us to remove 15 – 25% of duplicate entities in Group 1 (*work*, *expression*,*manifestation*). Furthermore, the set of Group 2 entities (*person*, *corporate body*) initially contains a fair amount of duplicates (respectively 70% and 80%). As a consequence, the dataset is cleansed, and the data processing (query, search) is accelerated.

Entity type	# of Entities before merging	# of Entities after merging	Ratio
Work	564379	422475	25%
Expression	451057	384954	15%
Manifestation	562838	465211	18%
Person	689957	207536	70%
Corporate Body	189532	38701	80%

Table 2
Merging Results

8.2.2. Quantitative Completeness, Redundancy and Extension

Next, we apply the quantitative metrics defined in Section 7: completeness, redundancy and extension. Table 3 shows the values achieved for properties, i.e., MARC control fields and datafields. We first observe that our format does not lose much data (both completeness values above 90%). The reason is because we sometimes change the datafield tags during the conversion back to our alternative MARC representation. As an example, 700 fields in the source record may include both the title of a work and the name of the person that is the author. Since we interpret these as separate entities we use 740 fields instead for the title in the work record.

Dealing with the amount of redundancies, it appears that the format tends to duplicate around 25% of the control and datafields. As explained in Section 5.2, the hierarchical representation involves redundancies. Our hybrid algorithm may select this representation for relationship types which have not been specified in the set of constraint, thus leading to duplicates. However, we insist on the fact that these duplicates could be easily deleted with a simple script after the conversion back to MARC.

Finally, we check the amount of semantic information which has been added. Namely, 8% of the datafields have been enriched with regards to the initial ones. Note that this amount only includes enriched fields as a result of the correction process (and not the types added as semantic elements). As expected, control fields have not been extended since their main purpose is to provide general information about a record.

To summarize, our format ensures a correct completeness. It does not guarantee a minimum number

[12] `http://dbpedia.org/lookup` (Feb. 2011)
[13] `wiki.freebase.com/wiki/Search` (Feb. 2011)
[14] DBpedia only, we did not query Freebase with MLQ language.
[15] 2.4 million links in November 2008

of redundancies but the duplicate properties can be removed with a post-conversion process. The semantic enrichment is also propagated back to the converted MARC records.

	Property	
	Control Fields	**Data Fields**
Completeness	97%	93%
Redundancy	25%	28%
Extension	0%	8%

Table 3

Quantitative Completeness, Redundancy and Extension for the NORBOK Collection

8.3. Qualitative Evaluation

The quantitative metrics do not provide insight about the quality. In this section, we present results on qualitative completeness and qualitative extension.

8.3.1. Qualitative Completeness

We have computed the following results for the NORBOK collection based on the qualitative completeness metric described in Section 7.1. Recall that this metric measures the amount of entities we have been able to interpret during the conversion process. In this part of the experiment, we focus on the two most interesting entities, i.e., *work* and *person*. Out of total "818,249" person entities in the original records identified using their key properties, we have been able to interpret "689,957", thus achieving 84% qualitative completeness. For the *work* entity, 88% of the fields that potentially identify works have been processed. These results are affected by the quality of the data and the rules that we have been able to create for this dataset.

8.3.2. Qualitative Extension

In this section, we evaluate the quality of the semantic enrichment process, namely the discovery of a semantic element for an added entry. Recall that the semantic element corresponds to the type of an entity, which is selected by querying different knowledge bases using the techniques describes in Section 6.1. It is not possible to manually check the discovered entity for the whole collection. Thus, we have randomly chosen 800 records for evaluation. These records contain 682 added entries for which we search for a semantic element. FRBR-ML computes a score for all entities and the one with the smallest score is selected. In this evaluation, we have ranked these entities and presented the top-3 candidate matches for validation (including a manual search on the knowledge bases for the entry value when needed). This validation step was performed by 8 people from our research group, which means that they have to check all proposed LOD entities and decide whether it corresponds to the given work (based on available information, such as creators, titles, summaries, or types). If none of the proposed entities is correct, participants validated the work by manually searching DBpedia and Freebase. This manual validation forms a ground truth for the 682 records, based on which we are able to compute quality results of our approach.

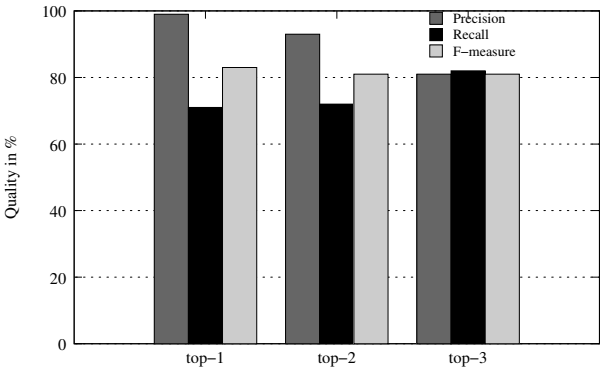

Fig. 10. Quality Results of the Semantic Process by Top-K

Almost half of the added entries do not have a corresponding entity in DBpedia or Freebase. Indeed, these added entries may refer to works or persons which are not popular enough to have a corresponding entity in the semantic knowledge bases. The remaining 343 works have at least one corresponding entity. We want to demonstrate that our semantic process identifies in most cases the correct entity at rank 1. Therefore, we compute the quality in terms of precision, recall and f-measure, as discussed in [17]. Applied to our context, *precision* represents the percentage of correctly identified entities among those discovered. On the other hand, *recall* stands for the percentage of entities correctly identified by our approach with respect to the total number of correct entities (based on ground truth). *F-measure* is a trade-off between precision and recall. Figure 10 depicts the quality obtained by our approach at top-1, top-2 and top-3. i.e., top-2 means that we consider entities which are ranked first and second by our semantic approach. For instance, top-1 results were obtained from this raw data: 155 true positives (correctly linked), 2 false positives (incorrectly linked), and 64 false negatives (not linked but should

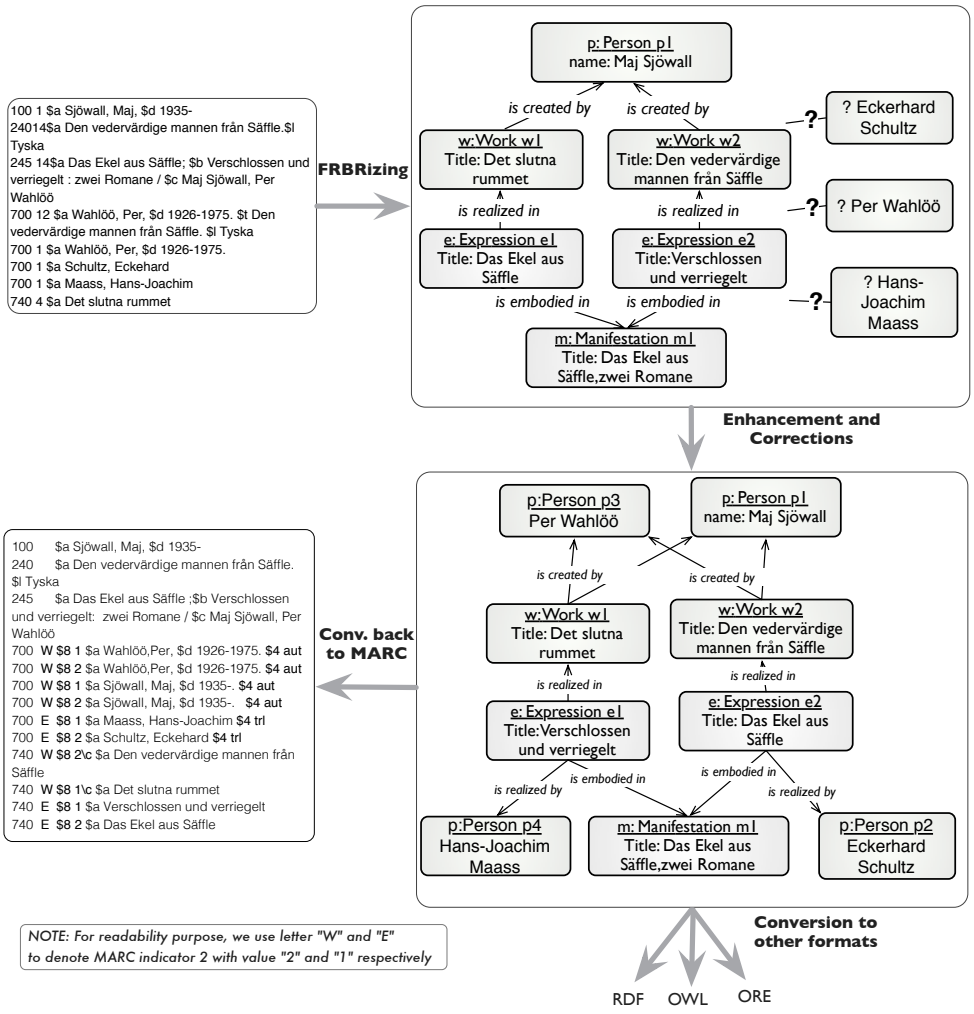

Fig. 11. An illustration of how the use case is solved.

have been). Thus, we achieve at top-1 99% precision score (155/157) and recall score of 71% (155/219).

We note that the precision at top-1 is close to 100%, which indicates that our approach does not discover too many incorrect entities. However, we miss some entities during the discovery process (recall equal to 71%). When considering the second and third ranked entities as well, we observe that more correct entities are discovered (recall values reaching 72% and 82%), but to the detriment of precision (decrease to 93% and 81%). As our semantic process aims at enriching records, it should ensure that we do not add too many incorrect semantic elements. In that case, our approach fulfills this goal since the first ranked entity selected by FRBR-ML is in most cases the correct one.

8.4. Solving a Complex Use Case

The qualitative extension only evaluates the quality of the semantic elements but it does not deal with the correction of the original ambiguous records. In Section 4.1, we presented a simple and a complex use case. The simple use case is tackled relatively easily because only one FRBR entity of each type is present in the record. However, the complex use case requires more effort to be solved. Figure 11 depicts the different transformations applied to the original MARC record up to the enriched one obtained by using our approach. The top part of the figure shows the initial MARC record and its FRBRized representation from the FRBRizer tool. We notice that both of them suffer from the same problems, i.e., the translators (Ecker-

hard Schultz, and Hans- Joachim Maass) and the second creator (Per Wahlöö) are not included in the output of the transformation because it is not clear how they are related to the entities found in the record.

The bottom right part of the figure illustrates the FRBR-ML based representation in which the missing semantic information about persons is enhanced and corrected. For instance, "Hans-Joachim Maass", "Eckerhard Schultz" and "Per Wahlöö" have been identified as *persons* by the knowledge-based matching method (Section 6.1.2). The second problem is tackled using the correction method (Section 6.2). For example, "Hans-Joachim Maass" is linked to the German expression "Verschlossen und verriegelt" that he has translated. The local collection lookup did not return any match for the expression title but querying $z3950.libris.kb.se$ with the query ""$find\ @attrset\ bib–1\ @attr\ 1\ =\ 4\ Verschlossen\ und\ verriegelt$" enabled to discover the correct relationship. On the other hand, the relationship between "Per Wahlöö" and "Det slutna rummet" was found during the intra-collection search since there is a record which has only this work. From this FRBR-ML format, it is possible to convert to RDF, OWL, ORE and back to MARC.

The bottom-left part of the figure shows the results of transformation from the FRBR-ML to MARC. We notice that it is the corrected and enhanced version of the original record. Entities are grouped by the $8 *linking field*. In our example, "$8 1" groups the work "Det slutna rummet", the expression "Verschlossen und verriegelt", the creators "Per Wahlöö" and "Maj Sjöwall", and the translator "Hans-Joachim Maass". For the second work, we applied "$8 2" as linking field. Indicators[16] W and E were adopted to denote whether the entity is related to work or expression. As an example, "Per Wahlöö" is related to both works since both indicators are W. In addition, the correct $relator\ codes$ "$4 trl" for translators and "$4 aut" for creators are used to denote their roles.

This new representation is inspired by both UNIMARC and MARC 21. The separation between names and titles comes from UNIMARC format while the use of the $8 linking field is common in MARC 21. Thus, our format has been adapted to fulfill our requirements, it is still compatible with with the ISO MARC standard.

[16]MARC indicator 2

9. Conclusion and Future Work

Experience and user feedback in the library community has shown that the adoption of new semantic technologies is slow, mainly because traditional library catalogs are still employed with records stored in the MARC legacy format. Thus, we have presented in this article our FRBR-ML framework on transforming and enhancing legacy bibliographic information into a representation where the structure and semantics of the FRBR model is explicit. The format in FRBR-ML can be used as an intermediary format to easily transform from/to MARC, RDF/XML, OWL and ORE. By writing an appropriate converter, one may also convert to other popular formats such as Dublin Core or ONIX. The enrichment step in our conversion consists of different strategies to tackle issues related to the lack of semantics in MARC records and to the identification of basic relationships between entities. We have studied novel techniques for disambiguating obscure entries in original records, thus allowing to correct the initial data. In addition, we have designed new metrics to check the quantity and quality of the transformation.

The results of experiments are promising. The merging process effectively removes duplicate entities, thus reducing the size of the collection in our format. However, our format includes redundant properties, but these redundancies can be easily removed during transformation to other formats. Additionally, it ensures a very high rate of completeness while allowing to correct and enhance ambiguous records with semantic information. We have also demonstrated that this semantic enrichment minimizes the rate of potential incorrect information.

In the future, we foresee several opportunities to improve our work. We are interested in discovering complex relationships between entities. To fulfill this goal, we plan to use pattern matching between involved entities. Next, we intend to cooperate with the National Library of Norway to apply our format. The user feedback from these librarians should help us detect the potential weaknesses and advantages of our approach in real world settings. Although we present our approach in the context of MARC-based information and the FRBR model, the solution is a generic framework that can be deployed for other types of information migration as well.

Acknowledgement

We thank the National Library of Norway for providing us an access to the NORBOK - The National Bibliography of Norway.

References

[1] T. Aalberg. A process and tool for the conversion of marc records to a normalized frbr implementation. In *Proc. of Int. Conference on Asian Digital Libraries*, pages 283–292, 2006.

[2] T. Aalberg, O. Husby, and F. B. Haugen. A tool for converting from marc to frbr. In *Research and Advanced Technology for Digital Libraries*, volume 4172, pages 453–456. Springer, 2006.

[3] T. Aalberg and M. Žumer. Looking for Entities in Bibliographic Records. In *Proc. of Int. Conference on Asian Digital Libraries*, Bali, Indonesia, 2008. Springer.

[4] A. K. Amin, J. van Ossenbruggen, L. Hardman, and A. van Nispen. Understanding cultural heritage experts' information seeking needs. In *Proc. of ACM/IEEE Joint Conference on Digital Libraries*, pages 39–47, 2008.

[5] C. Batini, M. Lenzerini, and S. B. Navathe. A comparative analysis of methodologies for database schema integration. *ACM Comput. Surv.*, 18(4):323–364, 1986.

[6] R. Bennett, B. F. Lavoie, and E. T. O'Neill. The concept of a work in worldcat: An application of frbr. *Library Collections, Acquisitions, and Technical Services*, 27(1):45–59, 2003.

[7] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia – a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7:154–165, September 2009.

[8] P. L. Boeuf. FRBR and Further. *Cataloging & classification quarterly*, 32, 2001.

[9] J. Bowen. FRBR: coming soon to your library? *Library resources and technical services*, 49(3):175–188, 2005.

[10] G. Buchanan, J. Gow, A. Blandford, J. Rimmer, and C. Warwick. Representing aggregate works in the digital library. In *Proc. of ACM/IEEE Joint Conference on Digital Libraries*, pages 247–256. ACM, 2007.

[11] M. da Conceição Moraes Batista and A. C. Salgado. Information quality measurement in data integration schemas. In *QDB*, pages 61–72, 2007.

[12] O. Dameron, D. Rubin, and M. Musen. Challenges in converting frame-based ontology into owl: the foundational model of anatomy case-study. *AMIA An. Symp Proc*, pages 181–5, 2005.

[13] I. Davis, B. D'Arcus, and R. Newman. Expression of Core FRBR Concepts in RDF, 2005.

[14] L. Dempsey and R. Heery. Metadata: a current view of practice and issues. *Journal of Documentation*, 54(2):145–172, 1998.

[15] M. Doerr and P. LeBoeuf. FRBRoo Introduction. *ICOM-CIDOC, version 0.8.1*, 2007.

[16] G. Dunsire. FRBRer model. `http://metadataregistry.org/schema/show/id/5.html`, February 2009.

[17] J. Euzenat and P. Shvaiko. *Ontology matching*. Springer, 2007.

[18] M. Franceschet, D. Gubiani, A. Montanari, and C. Piazza. From entity relationship to xml schema: A graph-theoretic approach. In *XSym*, pages 165–179, 2009.

[19] S. Geva, J. Kamps, and A. Trotman. Focused retrieval and evaluation, inex. In *INEX*, volume 6203. Springer, 2010.

[20] T. Health. LOD. *http://linkeddata.org/*, 2010.

[21] K. Hegna and E. Murtomaa. Data mining MARC to find: FRBR? In *68th IFLA General Conference and Council*, Glasgow, Scotland, 2002.

[22] T. B. Hickey, E. T. O'Neill, and J. Toves. Experiments with the IFLA Functional Requirements for Bibliographic Records (FRBR). *D-Lib Magazine*, 8(9), 2002.

[23] E. Hyvönen. Semantic portals for cultural heritage. In P. Bernus, J. Błażewics, G. Schmidt, M. Shaw, S. Staab, and R. Studer, editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 757–778. Springer, 2009.

[24] ISO. ISO 2709:2008. Information and documentation – Format for information exchange. `http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=41319`, 2008.

[25] ISO TC46/SC4. Information and Documentation : MarcXchange. ISO/CD25577, www.bs.dk/marcxchange, 2005.

[26] M. R. Jensen, T. H. Møller, and T. B. Pedersen. Converting xml dtds to uml diagrams for conceptual data integration. *Data and Knowledge Engineering*, 44(3):323–346, 2003.

[27] D. A. Koutsomitropoulos, G. D. Solomou, A. D. Alexopoulos, and T. S. Papatheodorou. Semantic web enabled digital repositories. *Int. Journal on Digital Libraries*, 10(4):179–199, 2009.

[28] D. A. Koutsomitropoulos, G. D. Solomou, and T. S. Papatheodorou. Metadata and semantics in digital object collections: A case-study on cidoc-crm and dublin core and a prototype implementation. *Journal of Dig Information*, 10(6), 2009.

[29] Library of Congress. MARC 21 Specifications for Records Structure, Character Sets and Exchange Media. *Network Development and MARC Standards Office*, January 2001.

[30] Library of Congress. Information and documentation – A reference ontology for the interchange of cultural heritage information. Technical report, , 2004.

[31] Library of Congress. Metadata Object Description Schema: MODS. Technical report, Library of Congress, 2004.

[32] Library of Congress. MARC 21 XML Schema. 2006.

[33] H. M. Á. Manguinhas, N. M. A. Freire, and J. L. B. Borbinha. Frbrization of marc records in multiple catalogs. In *Proc. of ACM/IEEE Joint Conference on Digital Libraries*, pages 225–234. ACM, 2010.

[34] K. Mcgrath and L. Bisko. Identifying FRBR Work-Level Data in MARC Bibliographic Records for Manifestations of Moving Images. *The Code4Lib Journal*, 1(5), December 2008.

[35] G. Pierra, J. C. Potier, and E. Sardet. From digital libraries to electronic catalogues for engineering and manufacturing. In *International Journal of Computer Applications in Technology (IJCAT*, pages 27–42, 2000.

[36] J. Riley. Enhancing Interoperability of FRBR-Based Metadata. In *International Conference on Dublin Core and Metadata Applications*, Pittsburgh, PA, USA, October 2010.

[37] H. Rode, P. Serdyukov, and D. Hiemstra. Combining document- and paragraph-based entity ranking. In *SIGIR*, pages 851–852, 2008.

[38] M. Rowe and F. Ciravegna. Data.dcs: Converting legacy data into linked data. In *Linked Data on the Web Workshop, World Wide Web Conference*, 2010.

[39] R. Tennant. *XML in Libraries*. Number 213p. Neal-Schuman Publishers, New York, 2002.

[40] The International Federation of Library Associations and Institutions. Functional requirements for bibliographic records. *UBCIM Publications - New Series Vol 19.*, 1998.

[41] A.-M. Vercoustre, J. A. Thom, and J. Pehcevski. Entity ranking in wikipedia. In *SAC*, pages 1101–1106, 2008.

[42] E. Voorhees and D. Harman. Trec experiment and evaluation in information retrieval. The MIT Press, USA, 2005.