LODLI: Linked Open Data Back-End System for Libraries Interoperability

Miriam Allalouf[1], Alex Artyomov[1] and Dalia Mendelsson[2]

[1] *Azrieli College of Engineering (JCE), Jerusalem, Israel*

[2] *The Hebrew University of Jerusalem, Jerusalem, Israel*

Keywords: Linked Open Data for Library Catalog Data, Mapping MARC to Ontologies.

Abstract: Linked Data principles offer significant advantages over current practices when publishing data. Linked Data allows library interoperability by linking to data from other organizations with authoritative data, which enriches library catalog-user search results. This paper describes LODLI, a Linked Open Data Back-End system that we designed and developed to enhance library catalog searches. We integrated our system with the Hebrew University library catalog, HUfind. While our platform can be used as is, it can also be customized by Linked Open Data providers that desire to convert their MARC records into Linked Data information library systems, making their data far more accessible. This research project faced the following challenges: finding the most efficient way to translate binary MARC into MARC records; mapping the MARC records into a variety of information models, such as Dublin Core, FRBR, RDA, OWL and FOAF, while selecting the most appropriate MARC field combinations; and providing links to resources in external datasets using a distance algorithm to identify string similarity. LODLI is a generic system to which additional ontologies can easily be added. We have demonstrated the system with two types of clients: FRBR visualization client and VIAF-extension client.

1 INTRODUCTION

Linked Data is a web-based method used for creating typed links between data from different sources that can be very diverse. The data sources can be databases at two organizations in different geographical locations, or heterogeneous systems within a given organization that historically haven't experienced easy interoperability at the data level. Linked Data (Berners-Lee, 2009; Bizer et al., 2009) refers to data published on the web in such a way that its meaning is explicitly defined and can therefore be processed by machines, which enables data from different sources to be connected and queried. Linked Data forms RDF graph located in a database that can be traversed to create a context for the described resources[1]. RDF predicates that share common domains are defined in ontologies.

Linked data principles makes data more accessible and useful by publishing it in a machine-readable format. For example, when a user agent queries a base to collect data about Venus, it should clearly understand the entity this data applies to —be it Venus (the planet) or Venus de Milo (the statue). Organizations including VIAF[2] (The Virtual International Authority File), WorldCat[3], and Wikipedia's databases (DBpedia and the newer Wikidata) collect data from varied resources on the web, process and store it in their Link Data database, and then reinsert it on the web.

Nowadays, libraries around the world desire to enjoy the benefits and advantages of linked data (Schilling, 2012; Coyle, 2010; Gonzales, 2014). The academic library catalog allows end users to search for and find the requested information. The Google-like search method influenced the way that library catalogs are being developed to offer a similar user experience (Emanuel, 2011), (Ramdeen and Hemminger, 2012). Presently, library catalogs offer academic resources based on the advances of information retrieval technologies (Merčun and Žumer, 2008; Tennant, 2005) that allow predictive search features (such as "Did you mean?"); user profile-aware content, such as tags, ratings, reviews, and comments; and a faceted classification. The new concept was developed only

[1] http://www.w3.org/TR/2004/REC-rdf-primer-20040210/

[2] http://viaf.org/

[3] http://www.worldcat.org/

Allalouf, M., Artyomov, A. and Mendelsson, D..
LODLI: Linked Open Data Back-End System for Libraries Interoperability.
In *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015) - Volume 2: KEOD*, pages 269-278
ISBN: 978-989-758-158-8

at the presentation layer of the Online Public Access Catalog (OPAC) and did not require changes in the Integrated Library Systems' (ILS) core. Most library catalogs, in order to supply basic functionality of resource description and discovery, describe their bibliographic records in MARC (Machine-Readable Cataloging). The extractable information from a typical MARC record is limited and fails to meet growing user expectations and needs in terms of sharing library collections and external resources access. Given world-wide emerging data collection and web publishing, it is essential and beneficial to apply Linked Data principles to information library systems.

We are not aware of any off-the-shelf Linked Data back-end systems for individual libraries. We therefore designed and developed LODLI (a Linked Data Back-End system). LODLI is a generic platform aimed at mapping MARC records to a Linked-Data information model that will serve the libraries in two possible scenarios:

1. Publish their own collections using Linked Data's machine-readable APIs ;
2. Enrich library catalog results with supplementary information taken from remote organizations.

In the first scenario, a library has its own collections, stores them in the LODLI repository and publishes the collections using Linked Data's machine-readable APIs such that other libraries may enjoy them. The design of LODLI requires a preliminary research stage to define these client applications. Given that the data interoperability area is still emerging and the use cases are not fully known or resolved, we had to attempt to predict the types of client applications and then build LODLI in a way that would address all the cases. Hence, the research questions for this scenario are: 1) Which objects and typed-connections a client library may require the new repository to have? and 2) Which Linked-Data information model and ontology, the MARC records will be converted to?

Our research observed that FRBR (Functional Requirements for Bibliographic Records) (Tillett and Cristian, 2009) is an appropriate information model to serve many types of client libraries that wish to find the semantic details of an item. FRBR applies a new model to the metadata of information objects in place of the flat record concept underlying the current MARC format. Linked Data concepts provide similar hierarchy and conform to the FRBR information model. The FRBR visualization client and the details behind the specific mapping of MARC fields to FRBR categories are described in Section 2.1. We also mapped the MARC records to the properties associated with Dublin Core that is suitable to describe library resources such as books and digital media, FOAF and OWL. Each structure was observed and selected according to its suitability to common client applications and will be described in Section 2.

The second scenario consists of libraries that wish to enhance their catalog capabilities by storing and managing external links to other data sets that describe the same resource. One challenging research question is which repositories to access for this resource and how to find the intended one. There are organizations with a large set of data that was collected from smaller libraries. The goal is to access a higher-tiered repository. For example, VIAF is a focal point for authoritative information collected from all the national libraries and it makes sense for LODLI to access it when looking for an author's metadata. We have tried to determine the repository access route that will be common to many catalog discovery applications. We provided the implementation of one pattern to be detailed in Section 2.2 where the client accesses first WorldCat and then VIAF.

Moreover, each organization that collects metadata for the same item, identifies it differently using their own ID. It will be convenient to access a higher-tiered repository assuming the ID is already known and we can correctly identify the required information. Thus, if LODLI is owned and customized by a national library that is aware of VIAF ID per each item, the access to the correct item will be easy. Otherwise, LODLI applies a fuzzy matching algorithm to ensure the remotely accessed item is the intended one.

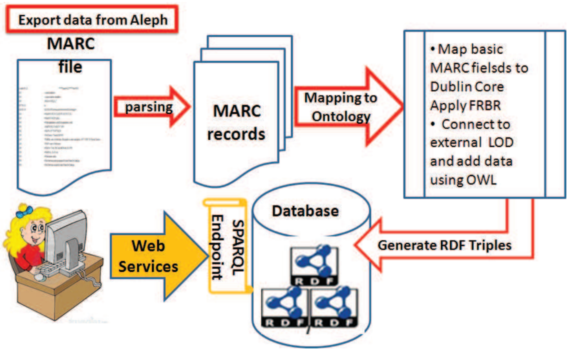


Figure 1: LODLI Flow.

Figure 1 is a LODLI dataflow diagram. The red arrows show the 3-stage generation and population of the Linked Data storage process, which occurs upon database creation and the addition of new database updates. Note that libraries that use LODLI to enrich their catalog using remote information have to covert their MARC records and store them in LODLI, as well. The orange arrow at the bottom represents a library catalog user who triggers a LODLI client, which provides access to external datasets. LODLI offers an end-to-end solution for a Linked Open Data publishing system and its architecture design followed the requirements for modularity and possible future extensions of additional information models, ontologies and discovery routes.

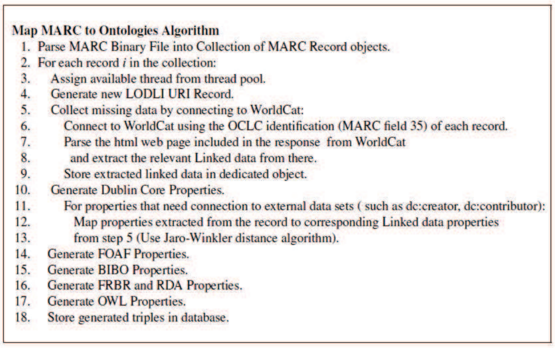
HUfind (http://hufind.huji.ac.il/) is the modern catalog used by the Hebrew University of Jerusalem, which serves its eight libraries and various faculties on the university's four campuses. The university's Library Authority, which is responsible for managing information technology for all university libraries, expressed the desire for enhanced catalog capabilities, which would enable them to access other libraries across the globe via a linked data system. LODLI's client were designed to be integrated with HUfind.

Section 2 characterizes the applications that will access LODLI and describes the mapping process of MARC records to an ontology-annotated RDF database. Section 3 describes the LODLI architecture. Section 4 presents the experimental setup and system evaluation. Section 5 provides related works in the field.

2 CLIENTS AND MARC MAPPING

To allow many types of client library applications, the LODLI system utilizes a full set of Linked Data principles, offering a platform to libraries that desire a Linked Open Data web system. The platform receives a file containing one or several MARC records as an input, extracts all records from the file, applies an FRBR structure to each record, maps the MARC records to the properties associated with Dublin Core, FOAF, OWL models, connects them to external datasets for additional information, and then stores them as RDF triples in a dedicated database. The pseudocode of the main algorithm performed by LODLI is depicted in Figure 2.

In this paper, we described two variants of LODLI clients: LODLI FRBR visualization client (Section 2.1) and the LODLI VIAF WorldCat client (Section 2.2). Both were implemented using standard client-side language, such as JavaScript, HTML and CSS. Cross-browser clients were tested and found to be fully functional in all commonly used browsers, such as Google Chrome, Mozilla Firefox, Opera, and Internet Explorer (starting version 8).



```
Map MARC to Ontologies Algorithm
 1. Parse MARC Binary File into Collection of MARC Record objects.
 2. For each record i in the collection:
 3.     Assign available thread from thread pool.
 4.     Generate new LODLI URI Record.
 5.     Collect missing data by connecting to WorldCat:
 6.         Connect to WorldCat using the OCLC identification (MARC field 35) of each record.
 7.         Parse the html web page included in the response  from WorldCat
 8.           and extract the relevant Linked data from there.
 9.         Store extracted linked data in dedicated object.
10.     Generate Dublin Core Properties.
11.         For properties that need connection to external data sets ( such as dc:creator, dc:contributor):
12.             Map properties extracted from the record to corresponding Linked data properties
13.             from step 5 (Use Jaro-Winkler distance algorithm).
14.     Generate FOAF Properties.
15.     Generate BIBO Properties.
16.     Generate FRBR and RDA Properties.
17.     Generate OWL Properties.
18.     Store generated triples in database.
```

Figure 2: Main Algorithm for MARC Parsing and RDF Generation.

2.1 FRBR Visualization Client

FRBR is a 1998 standard that was created by the International Federation of the Library Associations and Institutions (IFLA) to reconstruct catalog databases and to reflect the conceptual structure of information resources (Tillett and Cristian, 2009). FRBR applies a new model to the metadata of information objects replacing the flat record concept underlying the current MARC format. The FRBR model belongs to the family of entity-relationship models and contains four levels of representation:

Work. A distinct intellectual or artistic creation, such as a URI representing the title ”The Tragedy of Hamlet, Prince of Denmark”.

Expression. The intellectual or artistic realization of a work, such as a URI representing the English-language version of Hamlet.

Manifestation. The physical embodiment of an expression of a work. For example a URI representing the publisher of Hamlet.

Item. The book itself-e.g., a copy of Hamlet.

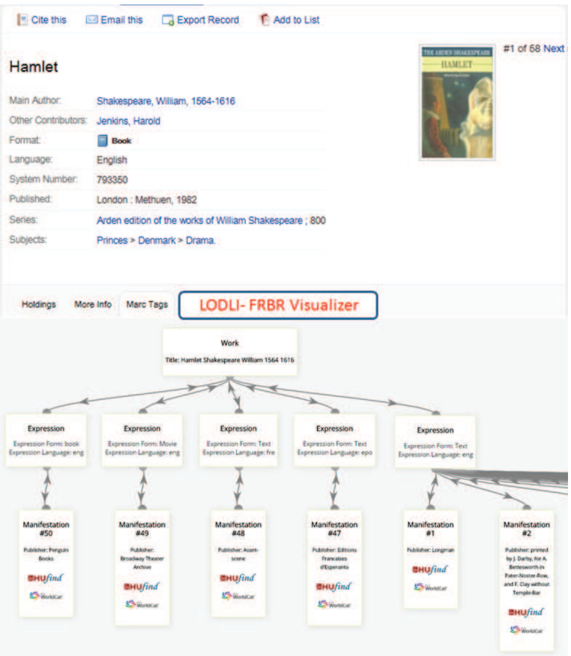
The use of FRBR to describe the data in the library offers a better user experience and superior resource discovery. For example, user searches for ”The Tragedy of Hamlet, Prince of Denmark” in the library site result in a list with the various expressions and manifestations of Hamlet in the library, such as a movie based on the play or translations into different languages.

For HUJI's library catalog, HUfind, we developed an FRBR visualization client. Figure 3 displays a Hamlet MARC record in the library identified by a given id. Clicking[4] on the LODLI FRBR Visualization button (in red) displays an interactive FRBR graph containing the three levels of the FRBR structure. The manifestation level displays hyperlinks to record pages in HUfind and WorldCat.

LODLI maps the MARC fields to FRBR-annotated RDF triples. We followed the Library of Congress mapping choice of MARC (Delsey, 2002),

[4] This client is ready and expected to be integrated with HUfind soon

where attributes associated with the FRBR entity *work* appear in the heading and title MARC fields. Attributes of the FRBR entity *expression* are recorded in textual form in a number of heading and title subfields as well as in certain material, physical and/or note fields. Attributes of *expression* are recorded as fixed length data elements, and attributes of the *manifestation* tend to be concentrated in the numbers, title, title-related fields, edition, and imprint fields.



Hamlet

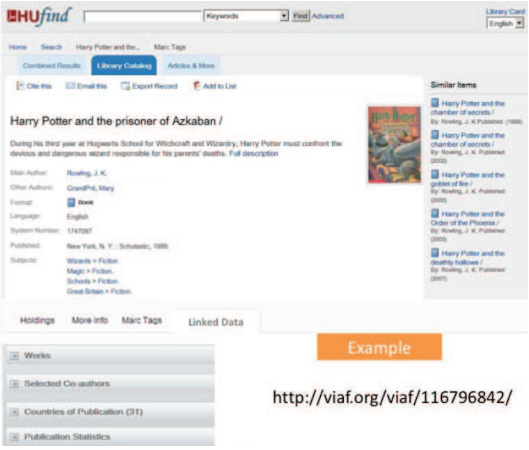
Main Author: Shakespeare, William, 1564-1616
Other Contributors: Jenkins, Harold
Format: Book
Language: English
System Number: 793350
Published: London : Methuen, 1982
Series: Arden edition of the works of William Shakespeare ; 800
Subjects: Princes > Denmark > Drama.

Figure 3: An FRBR visualization client integrated with HUfind.

2.2 VIAF and WorldCat Client

Figure 4[5] depicts an example for a client application that enriches HUfind user's search result using the LODLI system in the HUJI catalog. The upper part of the page in Figure 4 is a regular HUfind's presentation for the item Harry Potter and the Prisoner from Azkaban. Clicking on the Linked Data tab at the center of the page triggers our LODLI client implementation, which, in turn, accesses the link http://www.viaf.org/viaf/116846492/, which is a VIAF Authority File for J.K. Rowling the author of Harry Potter . As a result, a drop-down menu appears below the tabs, displaying additional search options. This Linked Data tab provides access to 18 national libraries around the world in addition to hundreds of other sources.

The Dublin Core Schema is a small set of vocabulary terms used to describe web resources (video, images, web pages, etc.), as well as physical resources such as books or CDs. The Library of Congress, the British Library, and Cornell University Library have conducted considerable work in the area of mapping MARC fields to corresponding Dublin Core properties and each of those libraries published a list of recommendations as to which mappings are preferable. On the basis of their recommendations, we used a combination of mappings in creating LODLI. The *creator* property describes the content of the resource and may refer to a person, an organization, or a service. We mapped *creator* by MARC fields 110, 111, 112. For the *contributor* property we followed the British Library recommendations that distinguishes between a person or an organization according to the MARC field and subfield. The Cornell University Library maps the *contributor* property by MARC fields 700, 710, 711 and 720. The Library of Congress suggested a slightly different mapping for the *contributor*, which is 700, 710, 711$e and 720$e. We chose the British Library recommendation that is more precise. The additional Dublin Core properties that are mapped in LODLI are summarized in Table 1.



Harry Potter and the prisoner of Azkaban /

During his third year at Hogwarts School for Witchcraft and Wizardry, Harry Potter must confront the devious and dangerous wizard responsible for his parents' deaths. Full description

Main Author: Rowling, J. K.
Other Authors: GrandPré, Mary
Format: Book
Language: English
System Number: 1747097
Published: New York, N. Y. : Scholastic, 1999.
Subjects: Wizards > Fiction
Magic > Fiction.
Schools > Fiction.
Great Britain > Fiction.

Harry Potter and the chamber of secrets / By: Rowling, J. K. Published: (1999)
Harry Potter and the chamber of secrets / By: Rowling, J. K. Published (2002)
Harry Potter and the goblet of fire / By: Rowling, J. K. Published (2000)
Harry Potter and the Order of the Phoenix / By: Rowling, J. K. Published (2003)
Harry Potter and the deathly hallows / By: Rowling, J. K. Published (2007)

http://viaf.org/viaf/116796842/

Figure 4: HUfind with LODLI Picture from VIAF.

A full Linked Open Data platform requires a connection to external Linked Data repository. Accordingly, some of our RDF triples were created by connecting our dataset with external datasets, such as VIAF and WorldCat. Connections were constructed by creating links between entities from our datasets (such as books and authors) to the same entities described in other datasets. For example, in our dataset, if "Harry Potter and the chamber of secrets" by J.K. Rowling is identified by http://hujilinkeddata/Record/HUJ001747095 and in WorldCat the same book it's identified by http://www.worldcat.org/oclc/318422828, by generating and storing the http://

[5] This client is ready and expected to be integrated with HUfind soon

Table 1: A summary of all the model properties mapped from MARC fields in LODLI.

Name	**Properties**
FRBR Functional Requirements for Bibliographic Records http://purl.org/vocab/frbr/core	*Work, Expression, Manifestation, RealizationOf, Realization, Embodiment, EmbodimentOf*
dc Dublin Core http://purl.org/dc/terms/1.1/	*contributor, coverage, creator, created, issued, description, format, identifier, relation, rights, source, subject, title, type, isFormatOf, hasFormat, isPartOf, haslanguage, publisher,isVersionOf, hasVersion, isReferencedBy, requires, Part, isReplacedBy, replaces, extent*
Bibo The Bibliographic Ontology http://purl.org/ontology/bibo	*Isbn, isbn10, isbn13, Conference*
foaf - Friend of a Friend http://xmlns.com/foaf/0.1/	*Person,Agent,Organization, Name*
OWL Web Ontology Language http://www.w3.org/2002/07/owl	*SameAs*
RDA Resource Description and Access http://rdvocab.info/Elements/	*IdentifierForTheWork, Title, IdentifierForTheExpression, LanguageOfExpression*

hujilinkeddata/Record/HUJ001747095 owl:SameAs http://www.worldcat.org/oclc/318422828 RDF triple we connect our dataset to the WorldCat.

HUJI's MARC record of Harry Potter (or any other item) does not include the VIAF ID of the author but it includes the WorldCat ID of the book. In order to connect HUJI's Authority entities (person names) to a VIAF object we used WorldCat linked data store. When the linked data is extracted from WorldCat, we fetch all the *creators*[6] and *contributors* properties of a particular book and then map them to the *creators* and *contributors* from our dataset using the distance algorithm for string matching. We had to use a string matching algorithm because in some records, the spelling of the names of creators and contributors is different in WorldCat from that in our dataset. Additionally, some authors that appear in our dataset as *creators* appear in WorldCat as *contributors*, and vice versa. By using the distance algorithm that calculates a similarity score for any two strings (e.g., authors names), we can perform mapping accurately thereby minimizing the mapping mistakes. Note, that if the VIAF ID is included in the library records, LODLI will insert it to its database and the string matching process will not be required any more. For the current client that wishes to find more information about an author of a certain item, we access first WorldCat, then VIAF and finally apply string matching. Other applications will require other discovery paths.

Another example of a client application we believe will serve library catalog users is an application that recommends authors. In this application the user provides the name of an author he likes and the application recommends similar authors. Author Similarity is based on the topics associated with the books written by the given author. These topics can be found in the MARC records. To retrieve the information LODLI will access VIAF for more authors and WorldCat for their books and topics. This work has not terminated yet and we have only preliminary results.

A **summary** of all the model properties mapped from MARC fields in LODLI appears in Table 1. The ontology name along with its namespace URI appears in the first column. All the properties that were implemented in LODLI appears in the third column.

3 SYSTEM ARCHITECTURE

LODLI's platform is depicted in Figure 5 and consists of Parsing Module, which parses MARC files into an intermediate collection of marc records that can serve as inputs to other modules; RDF Generation Module, which builds valid RDF triples from the MARC record data, Interconnectivity Module, which enriches the dataset by interconnecting it with external sources; and Database Access Module, which manages all interactions using the Virtuoso database.

The interaction between all the modules is done through explicitly defined interfaces, such that each module can be easily replaced, extended or modified without affecting other parts of the platform. We adopted an Inversion of Control(IoC) programming paradigm to create loosely coupled and high cohesion components. The binding process between different modules was achieved using a dependency injection pattern that enables performing coupling at the runtime. For error logging, we used the aspect-oriented programming paradigm, implemented through Spring Framework[7]. In order to improve the system's per-

[6] *creator* and *contributor* are Dublin Core properties that describe authors and contributors

[7] http://spring.io/

formance, we used the boss/workers multithreaded model, implemented with a configurable thread pool. By adding multithreaded processing, operational latency of the system was reduced by more than 14 times on average, as can be seen in Section 4.

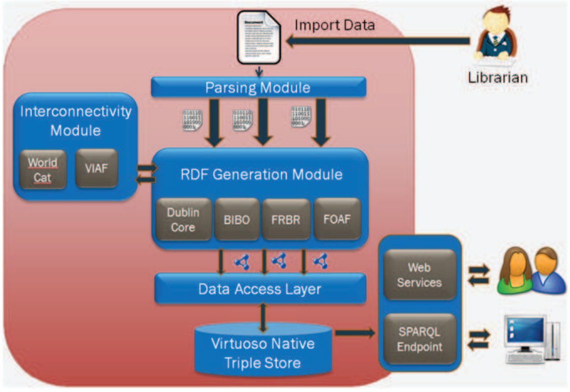


Figure 5: LODLI System Architecture.

Parsing Module receives a file that contains MARC records stored in an ISO2709 binary format as an input and then parses it into a collection of Java objects. Each of these objects contains a single record's metadata arranged into a set of nested objects and data structures that provide effective ways to access and manipulate the objects' properties. This module code is based on MARC4J[8], an open source Java toolkit.

RDF Generation Module is responsible for creating RDF triples from MARC-record properties using the mapping rules described in Section 2. The module consists of a collection of Java classes each of which is responsible for creating RDFs for a specific ontology. Each class encapsulates all requisite logic and business rules for fetching ontology-related data from a single MARC record. We mapped the MARC records into properties of Dublin Core, FOAF, OWL and other information models, as described in Table 1. All ontology classes implement a single common IOntology interface. IOntology interface uses a facade design pattern for simplifying the creation of all ontology properties. This architecture allows us to dynamically introduce new ontologies into the platform without changing any exiting code.

Interconnectivity Module handles the discovery routes and establishes connections between our dataset and external ones, such as VIAF and WorldCat. Connections are made by creating links between entities from external datasets, such as books and authors in other datasets. The functionality of this module was described in Section 2.2.

Database Access Layer deals with storing and retrieving RDF triples from our database. The layer is implemented using a repository design pattern, exposing an interface IRepository to all its consumers. Any database used as our storage platform will have to implement IRepository interface. Applying this architecture enables the decoupling of the actual database from other modules. Thus, the choice as to which database to utilize, and its implementation, will have no effect on other parts of the system. For LODLI we used the open source edition of Virtuoso Universal Server[9] that exposes SPARQL endpoint that can be used by clients to query the platform-generated dataset; demonstrated the best performance in most of the benchmark results, according to Berlin SPARQL Benchmark[10]; and supports both horizontal and vertical scaling. Horizontal scaling, including clustering and replication, are available in the commercial edition. All the RDF triples generated by LODLI are stored in Virtuoso RDF Quad Store and accessed using a Virtuoso Jena RDF Data Provider[11].

4 LODLI RESULTS AND EVALUATION

A Visualized LOD example consists of the generated RDF triples for the ”Hamlet” MARC record is presented in Figure 6.

The blue ellipses in the figure contain string literals that were extracted from MARC fields, or from a remote resource, and weren't a reference point (i.e., they are leaves in the RDF graph). Cyan rectangles represent LODLI complex entities containing additional properties capable of being dereferenced (e.g., http://hujilinkeddata/Record/HUJ00257250, which represents the Hamlet MARC record that appears in the center). Orange rectangles represent corresponding entities in external datasets, such as VIAF and WorldCat. The generated content of LODLI's database enables queries such as: 1) ”Given the work Hamlet, what are the different expressions in English?” where expression means book or video and it is searched for in a variety of repositories; or 2) ”What is the full set of the associated topics for the book title Hamlet?”.

In this work we identified several types of interesting applications that can be employed only when using our system. Consequently, we addressed the first

[8] https://github.com/marc4j/marc4j

[9] Virtuoso Universal Server: http://virtuoso.openlinksw.com/

[10] http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/results/V7/#machine

[11] Jena: http://jena.apache.org/

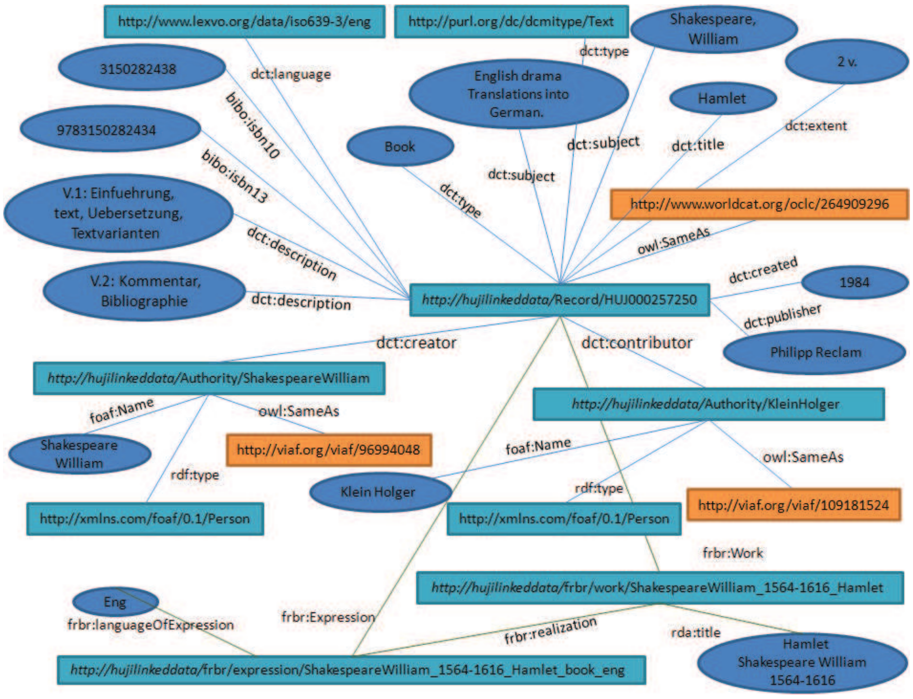

http://www.lexvo.org/data/iso639-3/eng
http://purl.org/dc/dcmitype/Text
Shakespeare, William
3150282438
dct:language
dct:type
2 v.
English drama Translations into German.
Hamlet
9783150282434
bibo:isbn10
Book
dct:subject
dct:title
dct:extent
bibo:isbn13
dct:subject
dct:type
http://www.worldcat.org/oclc/264909296
V.1: Einfuehrung, text, Uebersetzung, Textvarianten
dct:description
owl:SameAs
dct:created
1984
V.2: Kommentar, Bibliographie
dct:description
http://hujilinkeddata/Record/HUJ000257250
dct:publisher
dct:creator
dct:contributor
Philipp Reclam
http://hujilinkeddata/Authority/ShakespeareWilliam
http://hujilinkeddata/Authority/KleinHolger
foaf:Name
owl:SameAs
Shakespeare William
http://viaf.org/viaf/96994048
foaf:Name
owl:SameAs
rdf:type
Klein Holger
rdf:type
http://viaf.org/viaf/109181524
http://xmlns.com/foaf/0.1/Person
http://xmlns.com/foaf/0.1/Person
frbr:Work
http://hujilinkeddata/frbr/work/ShakespeareWilliam_1564-1616_Hamlet
Eng
frbr:languageOfExpression
frbr:Expression
frbr:realization
rda:title
Hamlet Shakespeare William 1564-1616
http://hujilinkeddata/frbr/expression/ShakespeareWilliam_1564-1616_Hamlet_book_eng

Figure 6: An example of the generated RDF triples for one MARC record (Hamlet). The links dct, foaf, rda, frbr and bibo are represented by the http://purl.org/dc/terms, http://xmlsns.com/foaf/0.1, http://rdvocab.info/Elements, http://purl.org/vocab/frbr/core# and bibo - http://purl.org/ontology/bibo URIs resptectively.

scenario research questions, namely ”Which objects and typed-connections a client library may require the new repository to have”? and ”Which Linked-Data information model and ontology, will the MARC records be converted to”?. Obviously, newer to come applications may require other information models or ontologies than those we have suggested. In our opinion, the design of an automatic tool to generate more RDF triples annotated with other ontologies should be deferred until more usage knowledge is obtained. However, LODLI is generic enough to allow for its extension with new structures.

Obtaining information from remote repositories raises the research question ”which repositories should be accessed for a resource and how to find the intended one?”. The problem comprises selecting the repository that contains the most metadata (”higher-tier” LOD) and finding the intended item in this repository by obtaining its ID or by performing fuzzy matching. We use the term ’discovery route’ for the list of the repositories, along with the appropriate ID that an application has to access in order to answer a specific query. For example, the discovery route for the VIAF client (described in 2.2) will be accessing WorldCat using WorldCat ID and then accessing VIAF using string matching. An important future research subject concerns the design of an automatic tool to maintain the discovery routes. This tool will identify the repository tiers and the existence of appropriate IDs to access the intended item or topic with regard to the selected LOD. Each type of application will be assigned an optimal discovery route that fits the asked queries in this application in terms of data volume and accuracy.

For **Performance Evaluation**, the LODLI platform was set up, integrated, and tested. We used a single server platform, but we're aware that any future production system will require multi-server architecture that can be vertically scaled to enhance performance and to tolerate faults. Hence, we chose to use Virtuoso because it is scalable. For the server we used an Intel Core i7-2600 CPU @ 3.40 GHz with 4 Cores, 8 Threads and 8MB Cache. The server contains 2 * 4 GB DDR3-1333MHz RAM and a 1.5 TB Seagate Barracuda SATA disk. The operating system was Windows 7 64 bit SP1.

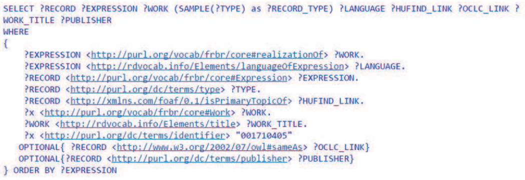
We used two benchmarks to evaluate LODLI's performance. The first tests the platform processing performance where new MARC records are parsed, translated into annotated RDF triples and inserted into the database. The second benchmark checks the query latency time experienced by a LODLI client.

The Parsing Benchmark was performed under two different dataset loads, a small number of MARC records (196 records) and a larger number of MARC records (47,527). It utilized thread pools of 1,4,16,32

Table 2: Results of parsing a small set of records and a larger set of records.

Number of threads in the thread pool	**Average RAM Memory Consumption**	**Average CPU Util˙ Percentage**	**Average processing time in milliseconds**	
The Parsing Benchmark with 196 MARC records				
1	87MB	13%	102249 ms	(1 minutes 42 sec)
4	88MB	13%	23934 ms	(24 seconds)
8	91MB	13%	14036 ms	(14.036 seconds)
16	91MB	15%	7231 ms	(7.231 seconds)
32	96MB	18%	5606 ms	
64	104MB	21%	3874 ms	
The Parsing Benchmark with 47,527 MARC records				
1	1.35GB	13%	3898842 ms	(1 hour 4 minutes 58 sec)
16	1.15GB	15%	291376 ms	(4 minutes 51 sec)
32	1.174GB	16%	229066 ms	(3 minutes 49 sec)
64	1.204GB	19%	215663 ms	(3 minutes 35 sec)

and 64 threads. Each thread was assigned several processing tasks where the parsing and RDF generation parts could be performed in parallel. However, the insertion of RDF triples into the database by each thread was conducted sequentially, which could lead to a bottleneck. We therefore evaluated the level of parallelism versus the database access rate under using a different number of threads. For each thread pool value we performed ten runs of parsing group records and measured maximum CPU, maximum memory consumption, and the time required to parse the records and store them in a database. The benchmark results appear in Table 2. As can be seen, when the size of the input is very small, the addition of parallel processing to the platform significantly improves the processing time. Operational latency was reduced by more than 14 times when 16 threads were used, and by more than 26 times for 64 threads. For low thread pool values there was no significant impact on CPU or memory consumption, since database insertions were operated serially. When a larger number of cases were involved, the results still demonstrated an improved processing time when increasing the number of threads. The resulting speedup was smaller than in the first case. It was improved by 16 times for 16 threads relative to one thread. To estimate the rate Viruoso inserts triples, we have ran benchmarks to insert 1 Million RDF triples using 16 and 64 threads. The demonstrated averaged latency times were 25 minutes for 16 threads and 27 minutes (2 minutes more) for 64 thread.



```
SELECT ?RECORD ?EXPRESSION ?WORK (SAMPLE(?TYPE) as ?RECORD_TYPE) ?LANGUAGE ?HUFIND_LINK ?OCLC_LINK ?
WORK_TITLE ?PUBLISHER
WHERE
{
    ?EXPRESSION <http://purl.org/vocab/frbr/core#realizationOf> ?WORK.
    ?EXPRESSION <http://rdvocab.info/Elements/languageOfExpression> ?LANGUAGE.
    ?RECORD <http://purl.org/vocab/frbr/core#Expression> ?EXPRESSION.
    ?RECORD <http://purl.org/dc/terms/type> ?TYPE.
    ?RECORD <http://xmlns.com/foaf/0.1/isPrimaryTopicOf> ?HUFIND_LINK.
    ?x <http://purl.org/vocab/frbr/core#Work> ?WORK.
    ?WORK <http://rdvocab.info/Elements/title> ?WORK_TITLE.
    ?x <http://purl.org/dc/terms/identifier> "001710405"
   OPTIONAL{ ?RECORD <http://www.w3.org/2002/07/owl#sameAs> ?OCLC_LINK}
   OPTIONAL{?RECORD <http://purl.org/dc/terms/publisher> ?PUBLISHER}
} ORDER BY ?EXPRESSION
```

Figure 7: FRBR SPARQL Query.

The Query Benchmark gauges the latency time of a complex query by measuring platform response time when it is populated by 1,158,110 triples. The benchmark was performed by using the FRBR visualization client described in 2.1. Figure 7 presents the SPARQL query used by the client. Latency time was the average result achieved from ten runs. We found that the average response time for this query ranged between 20 to 35 milliseconds.

5 RELATED WORK

Libraries and other cultural institutions' are currently undergoing tumultuous change. The Google-like search method influenced the way that library catalogs are being developed to offer a similar user experience (Emanuel, 2011), (Ramdeen and Hemminger, 2012). Nowadays, libraries' catalogues offer academic resources based on the advances of the information retrieval technologies; they are trying to meet the readers' new expectations and enhance their experience by making library catalogues more user friendly, intuitive, and visually attractive (Tennant, 2005). The breakthrough of providing web-based online public access to library collections and resources was developed only at the presentation layer of the Online Public Access Catalogue (OPAC). The Integrated Library Systems' (ILS) core was not changed and the fact that its content is encoded in natural language rather than as data infers library data integration with the Web (Coyle, 2010; Gonzales, 2014). Library standards serve only the library community, and changes in library technology are often totally dependent on the expertise of vendors (Baker et al., 2011). Schilling (Schilling, 2012) provides an excellent survey of the issues dealing with the transformation of library metadata into Linked Library Data. Barbara Tillett, who undertook the task of leading the selection and implementation of the Library of Congress' first Integrated Library System (ILS) and is well known for her development of the FRBR model stated in (Tillett, 2011) ”Our online catalogs based on MARC are no more than electronic versions of card catalogs with similar linear displays of textual information.” ”

There is a high commonality between libraries' traditional information management and interests (including constructing vocabularies, describing properties of resources, identifying resources, exchanging and aggregating metadata) and Semantic Web technologies, such as Linked Data principles (Heery, 2004). Currently, the web consists of links between resources and rich social interaction, providing some serendipity in their search results, while the library catalog offers little beyond search and retrieve. Coyle and Bourg (Bourg, 2010) have discussed the serendipity factor that linked data may add to library catalog search results. Bowen (Bowen, 2010) points to the need to develop tools for transiting libraries' existing legacy data to linked data, and describes the eXtenstible Catalog (XC) schema for linked-data-based catalogs.

In order to enrich library catalogs with Linked Data, standard technical tools need to be created. Additionally, library-related ontologies and value vocabularies have to be modeled. Styles *et al.*(Styles et al., 2008) discusses the possibilities of representing the most prevalent form of MARC, MARC21, as RDF for the Semantic Web. Heath *et al.*(Bizer and Hearh, 2011) describe the set of Linked Data publishing mechanisms. The primary consideration in selecting a workflow for publishing Linked Data includes the nature of the input data, the data preparation format, and data storage. Our selection of RDF storage for LODLI was a natural one, given that the fact that MARC records are structured input data. XSL Transformation (XSLT)[12] is the most common way to convert data into RDF from XML, though it requires that the MARC record first be translated to XML. The Library of Congress FRBR Display Tool (Aalberg et al., 2006) uses MARC4J, an open source Java toolkit, to convert MARC records stored in an ISO2709 binary format, to MARCXML, and then uses XSLT style sheets to convert them into MODS (Metadata Object Description Schema). Some of the drawbacks associated with using XSLT for RDF transformation, in addition to being cumbersome, are huge memory consumption during the parsing process and performance degradation. Hence, in LODLI we avoided XSLT and parsed MARC files into Java objects using MARC4J.

6 CONCLUDING REMARKS

This paper describes the LODLI platform we developed with the aim of enabling library catalogs, such as that at the Hebrew University, to link to other resources such as WorldCat and VIAF, thereby providing access to larger amounts of information. As a result of our mapping to Dublin Core, FOAF and FRBR model, libraries will be able to display relationships between novels, translated works, and all editions in the catalog as well as connecting the metadata to other cultivated resources. This platform can easily be extended and modified to add new ontologies, offering the functionality of Linked Open Data for libraries that wish to contribute their datasets. Moreover, LODLI enables the development of many interesting client and library-aware ontologies.

ACKNOWLEDGEMENTS

We thank Talia Cohen for running performance tests while working on the ”Author Recommendation” tool and for helpful discussions.

REFERENCES

Aalberg, T., Haugen, F. B., and Husby, O. (2006). A tool for converting from marc to frbr. *Research and Advanced Technology for Digital Libraries, ECDL 2006*.

Baker, T., Berm, E., Coyle, K., Dunsire, G., Isaac, A., Murray, P., Panzer, M., Schneider, J., Singer, R., Summers, E., Waites, W.and Young, J., and Zeng, M. (2011). Library linked data incubator group final report. www.w3.org/2005/Incubator/lld/XGR-lld-20111025 (Accessed August 30, 2015).

Berners-Lee, T. (2009). Linked data. in design issues. www.w3.org/DesignIssues/LinkedData.html, Accessed August 30, 2015.

Bizer, C. and Hearh, T. (2011). Linked data: Evolving the web into a global data space. *book*, 7(3).

Bizer, C., Heath, T., and Berners-Lee, T. (2009). Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 5:1–22.

Bourg, C. (2010). Linked data = rationalized serendipity. *Federal Librarian (blog)*.

[12] http://www.w3.org/TR/xslt20/

Bowen, J. (2010). Moving library metadata toward linked data: Opportunities provided by the extensible catalog. *International Conference on Dublin Core and Metadata Applications, DC-2010.*

Coyle, K. (2010). Understanding the semantic web: Bibliographic data and metadata. *Library Technology Reports.*, 46(1).

Delsey, T. (2002). Functional analysis of the marc 21 bibliographic and holdings formats. http://www.loc.gov/marc/marc-functional-analysis/source/analysis.pdf.

Emanuel, J. (2011). Usability of the vufind next-generation online catalog. *Information Technology and Libraries*, 30(1).

Gonzales, B. M. (2014). Linking libraries to the web: Linked data and the future of the bibliographic record. *Information Technologies and Libraries*, 33(4).

Heery, R. (2004). Metadata futures: Steps toward semantic interoperability. *Metadata in Practice.*

Merčun, T. and Žumer, M. (2008). New generation of catalogues for the new generation of users: A comparison of six library catalogues. Program.

Ramdeen, S. and Hemminger, B. M. (2012). A tale of two interfaces: How facets affect the library catalog search. *JASIST*, 63(4):702–715.

Schilling, V. (2012). Transforming library metadata into linked library data. *American Library Association*. http://www.ala.org/alcts/resources/org/cat/research/linked-data (Accessed August 30, 2015).

Styles, R., Ayers, D., and Shapir, N. (2008). Semantic marc, marc21 and the semantic web. Talk at the WWW 2008 Workshop: Linked Data on the Web (LDOW2008), April 22, 2008 in Beijing, China.

Tennant, R. (2005). Lipstick on a pig. *Library Journal*, 130(7).

Tillett, B. (2011). Keeping libraries relevant in the semantic web with resource description and access (rda). *Serials*, 24(3).

Tillett, B. and Cristian, A. (2009). Ifla cataloguing principles: the statement of international cataloguing principles (icp) and its glossary in 20 languages.