German Medical Data Sciences: A Learning Healthcare System
U. Hübner et al. (Eds.)

doi:10.3233/978-1-61499-896-9-40

Metadata Import from RDF to i2b2

Mark R. STÖHR[a,1], Raphael W. MAJEED[a] and Andreas GÜNTHER[a]
[a] *UGMLC, German Center for Lung Research (DZL), Justus-Liebig-University, Giessen, Germany*

Abstract. Metadata management is an important task in medical informatics and highly affects the gain out of existing health information data. Data Warehouse solutions like Informatics for Integrating Biology and the Bedside (i2b2) are common tools for identifying patient cohorts and analyzing collected clinical data while respecting patient privacy. The Resource Description Framework (RDF) is designed for highly interoperable ontology representation in various formats, facilitating ontology and metadata management. Our approach is to combine i2b2's and RDF's benefits by importing the easy-to-edit RDF ontology into the extensive-research-enabling i2b2 software. We do so by using a SPARQL Protocol and RDF Query Language (SPARQL) interface, that enables RDF data queries, and developing a java program, which then generates i2b2-specific SQL insert statements. To demonstrate our solution's feasibility, we transcribe our lung disease specific ontology to RDF and import it into our i2b2 data warehouse.

Keywords. Biological ontologies, metadata, organization and administration, automatic data processing, information systems

1. Introduction

In medical informatics, metadata management is an important and demanding task. It is indispensable for data harmonization, data integration, data quality management and data comparison. Applied on clinical data, these processes result in large pools of consolidated, corrected and annotated data, enabling large-scale clinical data analysis and trial patient recruitment [1].

The clinical data warehouse software i2b2 serves as storage system for clinical data as well as metadata and offers a reliant and effective tool to support clinical trials by either prospectively finding cohorts of patients fulfilling specific constraints or retrospectively making further use of already collected routine health care data. Established in 2004, 7 years later already 60 academic health care centers have adopted this software [2]. It is funded by the National Institutes of Health (NIH). A large community continuously enhances its features. For example, by developing solutions for challenges created by i2b2's founders at Harvard MIT Division of Health Sciences and Technology. From 2006 to 2012 there are 124 publications listed, enabled by these challenges [3]. From 2016/01 to 2017/10 PubMed lists 46 articles related to i2b2, which shows that i2b2 and its applications are of ongoing interest and usefulness to the research community.

[1] Corresponding Author, Mark R. Stöhr, UGMLC, Justus-Liebig-University, Klinikstraße 36, 35392 Gießen, Germany; E-mail: mark.stoehr@innere.med.uni-giessen.de.

RDF is a W3C standard and designed for ontology representation and described resources may be annotated with literals (e.g. labels or codes) and connected to other resources for large knowledge-graphs. It is commonly used in many fields like content management, content discovery, data integration, semantic annotation and schema mapping. Several use cases show Health Care applications' clear dependency on these fields [4]. RDF statements consist of triples < Subject, Predicate, Object >, which describe the subject in more detail, either through literal attributes or through relations to other resources. RDF has a variety of representations, e.g. N3, Turtle, JSON or XML [5–7].

For research (meta)data, FAIR Data Principles offer a measurable set of principles to improve data quality in terms of data being Findable, Accessible, Interoperable and Reusable. Many implementations show how RDF serves as a reliant component for fulfilling these principles [8]. One may enrich RDF resources by referencing concepts from clinical ontologies like SNOMED-CT or LOINC through their globally unique identifier / code [9,10]. For this purpose, systems like Dublin Core (DC) or Simple Knowledge Organization System (SKOS) provide standardized relations and annotations [11,12]. By referencing common resources in a standardized way, RDF data becomes interoperable, findable and reusable.

Although several solutions exist for importing routine health care data [13], there are still very few tools for i2b2 ontology administration [14]. Particularly, there is no published solution for importing RDF metadata into i2b2.

This paper aims on developing a generic solution for metadata import of RDF metadata into an i2b2 database and prove its feasibility.

2. Methods

There are two effective ways for importing data into the i2b2 metadata structure: Either by working directly on the database or by using the provided web service API. The latter becomes less efficient for large-scale data import and is less flexible because of API restrictions. Therefore, we decided to query the database directly through SQL statements. To achieve this, we have to transform RDF resources into appropriate SQL statements, taking into account their relations (especially their hierarchy) as well as some annotations like labels and codes of other ontologies.

Four database tables are used for the i2b2 metadata representation: "i2b2", "table_access", "concept_dimension" and "modifier_dimension" [15]. The tables "i2b2" and "table_access" contain information about the visual representation of concepts in the user interface. This is where labels and descriptions are stored. The "concept_dimension" and "modifier_dimension" tables contain coding information. They are used internally for execution of user queries. Modifiers are implemented in i2b2 since release 1.6.00 to enrich existing concepts (and when needed its sub-concepts) with additional specification options. I2b2 can only display concept trees, so we can only import RDF graphs without loops.

The W3 Consortium published SPARQL [16] as a standard for querying RDF. By using an engine with a SPARQL-interface, we can provide accessibility (see FAIR principles) of RDF resources. Jena Fuseki Server is a popular open source software designed for querying loaded RDF data via SPARQL interface. It supports a wide range of input formats like Turtle, XML or JSON.

For each concept or modifier, we have to query the RDF data for the concept's primary label, codes, as well as sub-concepts and applied modifiers. Then we add a row to the i2b2 table. In case of annotated codes, which are necessary for concepts and modifiers to be found and considered in the results, we also add a row to the respective "*_dimension" table. In case of a root concept, we also add a row to the "table_access" table.

Knowing how to query the RDF data and what the i2b2 database insert statements have to look like, we are able to formulate an algorithm that will generate all insert statements by successively gathering all required concept and modifier information from the RDF graph (Figure 1). This algorithm recursively runs through a method that, for each concept, queries all its sub-concepts, modifiers, notations and its preferred label, generates the proper SQL-statement and then continues with all sub-concepts. Visual attributes for the i2b2 web client (e.g. folder, leaf, hidden sub-concepts) are deduced by looking at the number of sub-concepts. For modifiers, the same procedure is applied.

The German Center for Lung Research (DZL) uses i2b2 as central data warehouse software, collecting data from various sources and merging them for large-scale retrospective data analysis, sample ordering and patient recruitment. The advantages of properly setup metadata has already been shown in the introduction. To evaluate our metadata import algorithm, we transcribe our DZL ontology to turtle N3 syntax and load it into a Jena Fuseki server, offering the SPARQL interface for querying the data. The DZL ontology was composed by lung researchers, medical documentalists and data managers using our Collaborative Metadata Repository (CoMetaR) editing framework [17].

For hierarchical ordering of concepts and modifiers we use the standard RDF predicates skos:topConceptOf/skos:hasTopConcept and skos:broader/skos:narrower. For modifier indication, we use rdf:hasPart/rdf:partOf. For labeling and description, we use skos:prefLabel and dc:description. Concept's and modifier's codes are indicated through skos:notation. Table 1 shows an example of entries in the postgres database used by i2b2. All software components (RDF files, Jena Fuseki Server, our java program and i2b2 software) run on the same Linux system.

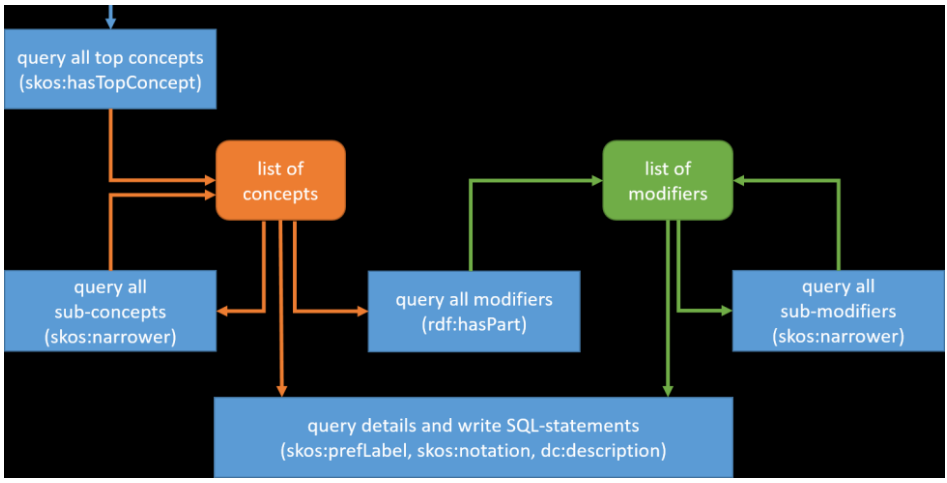

Figure 1. Visualization of the RDF-to-i2b2 transformation algorithm.

Table 1. Example entries in respective database tables for i2b2 (postgres).

Database table i2b2

c_fullname	c_name	m_applied_path
\i2b2\dzl:Dataset\dzl:Follow-up\	Follow-up	NULL
\i2b2\dzl:Dataset\dzl:Follow-up\dzl:Vitalstatus\	Vital sign	NULL
\L:8462-4\	Diastolic blood pressure	\i2b2\dzl:Dataset\dzl:Follow-up\dzl:Vitalstatus\S:75367002\%
\i2b2\dzl:Dataset\dzl:Follow-up\dzl:Vitalstatus\S:75367002\	Blood pressure	NULL

Database table concept_dimension

concept_path	name_char	concept_cd
\i2b2\dzl:Dataset\dzl:Follow-up\dzl:Vitalstatus\S:75367002\	Blood pressure	S:75367002

Database table modifier_dimension

modifier_path	name_char	modifier_cd
\L:8462-4\	Diastolic blood pressure	L:8462-4

3. Results

We developed a solution for integrating RDF ontologies into i2b2 databases by loading the data into an application that provides a SPARQL interface processing it through an algorithm, which queries the interface for RDF concepts recursively and generates SQL insert statements, before executing these SQL-statements. The ontology is accessible via SPARQL through https://data.dzl.de/fuseki/cometar_live/query. A visualization is available at https://data.dzl.de/cometar/.

3.1. Range and Performance of the Implementation

Our entire DZL ontology was successfully imported into i2b2. Currently it includes 653 concepts and 39 modifiers organized in four trees with 518 leafs. Out of all, 601 concepts and 32 modifiers are associated with a code. In total, the ontology is spread across 40 text files in turtle syntax (ttl). During transformation by our java algorithm, all insert statements are split up into two SQL files, since some have to be executed on i2b2metadata schema and some on i2b2demodata schema. Loading the ttl-files into Jena Fuseki Server takes less than one second. The java program for generating all i2b2 insert statements runs between 9 and 21 seconds. Executing the SQL files on PostgreSQL takes about two seconds.

4. Discussion

We developed a generic algorithm to import metadata from RDF to i2b2 and proved its feasibility. Although i2b2 can represent some basic elements of standard RDF vocabulary (e.g. SKOS), its capabilities are rather limited. For example, relations like skos:closeMatch and graphs with loops are not representable in i2b2. On the other hand, data type restrictions are supported, but our algorithm does not yet consider them. Editing

RDF ontologies in turtle N3 syntax by hand may prove difficult. In this case, RDF editing tools like Protégé may facilitate this task. Since execution time of our implementation lies under one minute, it is possible to perform live i2b2 ontology updates after editing the RDF ontology.

5. Conclusion

In order to combine the benefits of i2b2 and RDF, we investigated possibilities to transfer RDF metadata to i2b2 data warehouse systems. We found that import of hierarchical information, labels and codes is feasible and developed an algorithm for this task.

6. Conflict of Interest

The authors state that they have no conflict of interests.

References

[1] Q. Chong, A. Marwadi, K. Supekar, Y. Lee, Ontology Based Metadata Management in Medical Domains, *Journal of Research and Practice in Information Technology* **35(2)** (2003), 139–154.
[2] I.S. Kohane, S.E. Churchill, S.N. Murphy, A translational engine at the national scale: informatics for integrating biology and the bedside, *Journal of the American Medical Informatics Association : JAMIA* **19(2)** (2012), 181–185.
[3] Informatics for Integrating Biology and the Bedside, NLP Research Data Sets, 2017 [Last accessed: 10/27/2017], https://www.i2b2.org/NLP/DataSets/Publications.php.
[4] W3C Semantic Web, Semantic Web Case Studies and Use Cases, 2012 [Last accessed: 11/10/2017], https://www.w3.org/2001/sw/sweo/public/UseCases/.
[5] W3C Semantic Web, W3C Recommendation / RDF Documents and Syntaxes, 2014 [Last accessed: 11/10/2017], https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/Overview.html.
[6] W3C Semantic Web, Notation3 (N3): A readable RDF syntax, 2011 [Last accessed: 11/10/2017], https://www.w3.org/TeamSubmission/n3/.
[7] W3C Semantic Web, Turtle – Terse RDF Triple Language, 2014 [Last accessed: 11/10/2017], https://www.w3.org/TR/turtle/.
[8] M.D. Wilkinson, M. Dumontier, I.J. Aalbersberg, et al., The FAIR Guiding Principles for scientific data management and stewardship, *Scientific data* **3** (2016), 160018.
[9] Regenstrief Institute, LOINC home page, 2016 [Last accessed: 11/10/2017], http://www.regenstrief.org/resources/loinc/.
[10] College of American Pathologists, SNOMED Clinical Terms (SNOMED CT), 2017 [Last accessed: 11/10/2017], http://www.snomed.org.
[11] Dublin Core Metadata Initiative, Dublin Core, 2017 [Last accessed: 11/10/2017], http://dublincore.org/.
[12] W3C Semantic Web, Simple Knowledge Organization System (SKOS), 2017 [Last accessed: 11/10/2017], http://www.w3.org/standards/techs/skos.
[13] R.W. Majeed, R. Röhrig, Automated realtime data import for the i2b2 clinical data warehouse: introducing the HL7 ETL cell, *Studies in health technology and informatics* **180** (2012), 270–274.
[14] C.R. Bauer, T. Ganslandt, B. Baum, et al., Integrated Data Repository Toolkit (IDRT). A Suite of Programs to Facilitate Health Analytics on Heterogeneous Medical Data, *Methods of information in medicine* **55(2)** (2016), 125–135.
[15] Informatics for Integrating Biology and the Bedside, i2b2 software, 2017 [Last accessed: 11/10/2017], https://www.i2b2.org/software/index.html.
[16] W3C Semantic Web, SPARQL Protocol And RDF Query Language (SPARQL), 2017 [Last accessed: 11/10/2017], http://www.w3.org/standards/techs/sparql.
[17] M.R. Stöhr, R.W. Majeed, A. Günther, Using RDF and Git to Realize a Collaborative Metadata Repository, *Studies in health technology and informatics* **247** (2018), 556–560.