doi:10.3233/978-1-61499-808-2-42

On-The-Fly Query Translation Between i2b2 and Samply in the German Biobank Node (GBN) Prototypes

Sebastian MATE [a,1,*], Patric VORMSTEIN [b,*], Dennis KADIOGLU [a,c], Raphael W. MAJEED [d], Martin LABLANS [e], Hans-Ulrich PROKOSCH [a,f] and Holger STORF [b]
[a] *Medical Informatics, Univ. of Erlangen-Nürnberg, Erlangen, Germany*
[b] *Medical Informatics Group, University Hospital Frankfurt, Frankfurt, Germany; German Cancer Consortium (DKTK), partner site Frankfurt; and German Cancer Research Center (DKFZ), Heidelberg, Germany*
[c] *Institute of Medical Biostatistics, Epidemiology and Informatics, University Medical Center of the Johannes Gutenberg University Mainz, Mainz, Germany*
[d] *German Center for Lung Research, Justus-Liebig-University, Giessen, Germany*
[e] *Medical Informatics in Translational Oncology, German Cancer Research Center (DKFZ), Heidelberg, Germany*
[f] *Center for Medical Information and Communication, Erlangen University Hospital, Erlangen, Germany*

Abstract. Information retrieval is a major challenge in medical informatics. Various research projects have worked on this task in recent years on an institutional level by developing tools to integrate and retrieve information. However, when it comes down to querying such data across institutions, the challenge persists due to the high heterogeneity of data and differences in software systems. The German Biobank Node (GBN) project faced this challenge when trying to interconnect four biobanks to enable distributed queries for biospecimens. All biobanks had already established integrated data repositories, and some of them were already part of research networks. Instead of developing another software platform, GBN decided to form a bridge between these. This paper describes and discusses a core component from the GBN project, the OmniQuery library, which was implemented to enable on-the-fly query translation between heterogeneous research infrastructures.

Keywords. Data harmonization, federated search, biobanking, interoperability

1. Introduction

Medicine is a highly heterogeneous environment, and information integration is a recurring challenge in a medical informatics researcher's day-to-day business. Not only does the information content differ from the medical perspective, one has to deal with a plethora of different data sources. This is a major issue in networked research, where medical data has to be integrated in order to build large-scale data pools to enable the identification of patients with e.g. rare diseases or special types of cancer. In recent years, different approaches to address this problem were developed (an overview can

[1] Corresponding Author: **Sebastian.Mate@fau.de** [*] These authors contributed equally to this paper.

be found in [1]). However, when different toolsets are used across institutions, the issue of merging and retrieving medical information on a national or international level remains. Furthermore, the willingness of research partners to install additional software and to keep data available in a duplicated fashion is certainly limited, resulting in a need to interconnect already existing platforms.

The German Biobank Node project (GBN) [2], which is part of the *Biobanking and BioMolecular resources Research Infrastructure* (BBMRI) [3], aims to build a network of interconnected biobanks on a national level in Germany. The ultimate goal is to give researchers the ability to search for patients with certain diseases and their associated specimens. The project faced the above-mentioned issues of already exiting, heterogeneous research infrastructures. Aiming not to establish a dedicated GBN platform, the project decided to implement an approach for direct query message translation between the already existing research architectures. By injecting these translated messages into the other system, queries could be executed natively, without implementing an abstraction layer. In this paper, we describe a core component, the OmniQuery library, which enables this on-the-fly query translation between these different research architectures, and discuss the current limitations of our approach.

2. Methods

The GBN prototypes were implemented by four GBN partners and tested with their associated biobanks. Those are the DZL biobank (Giessen University Hospital), the Charité ZeBanC (Berlin), the biobank of the University Cancer Center Frankfurt (UCT Frankfurt) and the biobank of the Comprehensive Cancer Center Erlangen (CCC Erlangen-EMN). In recent years, research teams in Giessen and Erlangen developed i2b2-based data warehouses (DWHs) that are fed with sample data from their biobank systems [4,5]. Similarly, Berlin and Frankfurt established the commercial software CentraXX® (Kairos GmbH), which has been enhanced to serve as local DWHs within the German Cancer Consortium's (DKTK) bridgehead architecture "Samply" [6].

Against this background, GBN decided to support CentraXX® and i2b2 in its prototypes. The challenge was to establish a new hybrid architecture integrating both types of DWH. This required bridging between the already existing two architectures and enabling the translation of queries that were created with the user interface of one platform to the schema and ontology of the other platform. If the query was properly translated, it could be injected into the other architecture without the architecture recognizing that it originated from an external system. By that, it would not have been necessary to modify the original architectures on a large scale.

To meet the goal of executing shared queries (created at a central point within the network) on CentraXX® and i2b2, these have to be distributed to the biobanks at the different sites. To implement the GBN query distribution, we based our two prototypes on components from the Samply system and the "lightweight i2b2 server and client libraries" (li2b2, https://github.com/li2b2), a development from the AKTIN project in Giessen [7]. The Samply components, which were developed by the DKTK and used in Berlin and Frankfurt, allow distributed search among (in this case) CentraXX®-based biobanks. Likewise, the li2b2 components allow interconnecting i2b2-based DWHs.

Given that Samply and li2b2 were already available for query transport, GBN's major challenge was the query translation between CentraXX® and i2b2. To achieve a better understanding of both systems' query capabilities, we conducted a thorough

analysis of the two query formalisms used in Samply/CentraXX® and li2b2/i2b2. Both use XML-based query definitions to communicate the medical concepts used in a query, along with their logical relationships ("AND" and "OR"). They also share a similar approach for formulating and executing queries. The inclusion and exclusion criteria for feasibility queries are expressed by medical data elements, which are retrieved from a terminology service. Queries are then formulated by combining multiple data elements with Boolean logic. Data elements may describe the plain existence of a condition (e.g. "male gender") or a more detailed observation (e.g. blood glucose measurements). Furthermore, data elements can be grouped hierarchically on both systems. Depending on the data type of the data elements in a query, a numeric value including a comparator can be attached to the data element, but is also possible to simply check for the plain existence of a data element by omitting the numeric comparison. On both systems, it is also possible to exclude patients from the query result.

Table 1. Comparison between the two different query logics.

Feature	Samply/CentraXX®	li2b2/i2b2
Query Formalism	Full Boolean logic, models SQL formalism	Boolean logic in conjunctive normal form (CNF), models i2b2 formalism
Numeric Comparators	Greater or equal, greater than, equal, less or equal, less than, not equal, between, in (with slightly different naming)	
Check for Existance	"Is Not Null" comparison	Removal of numeric constraint
Non-Existance	"Is Null" comparison	Exclusion of query panel
Metadata	Samply MDR (ISO-11179-based [8])	i2b2 ontology (own model)
Grouping of Data Elements	Mono-hierarchical and catalogs (e.g. for LOINC, ICD-10)	Mono-hierarchical incl. utilization during query runtime (hierarchical subsumption)
Temporal Logic	None	Constrain by fixed dates, complex relative logic (sequence of events)
Other Features	None	Unit conversion, occurs, grouping (financial encounter, modifier)

Despite these similarities, and as outlined in Table 1, there are differences between both platforms, which need to be addressed in order to allow for translating query messages. Both systems express queries in XML syntax, but it is not possible to perform direct XML transformations due to the different types of Boolean logic used. However, this can be overcome by utilizing logic transformations based on rules on logic equivalence. The different naming of numeric comparators can be addressed by replacing the identifiers. Both systems support the "Check for Existence" query feature, which requires a data element to be available for a patient, independently of its value (for catalogs or numeric values). For numeric data elements, compatibility can be achieved by simply removing the numeric value in i2b2 queries or by replacing the comparator with "Is Not Null" in CentraXX® queries. Similarly, the opposite operator "Check for Non-Existance" can be translated in both directions easily. Because both systems utilize their metadata internally during query run-time, it is not necessary to perform syntactic translations related to metadata, except for concept mapping. This can be addressed by providing 1:1 mappings. Similarly, the conceptual subsumption feature of i2b2 can be replicated in CentraXX® by providing 1:n mappings.

3. Results

This analysis enabled us to implement OmniQuery, a Java library, which uses Plain Old Java Objects to hold content and logic of "generic" feasibility queries. Its instantiated objects represent a tree-like structure and are composed out of three main classes. An *OmniQuery* class acts as a root node, which can contain an unlimited number of child nodes, a *LogicNode* class represents logical associations as a child node and a *ConstraintNode* class defines the actual constraints as leaves of the tree structure. Furthermore, enumerations for marking *LogicNodes* and *ConstraintNodes* such as AND, OR, EQUALS, etc. have been integrated. This approach enables performing arbitrary, tree-related algorithms on the data structure and easily manipulating the structure such as changing ancestor and children of a given node. This object structure allows to represent the common features of both Samply and i2b2 queries. To perform the logic transformations, we integrated the open source library AIMA3e (https://github.com/aimacode/aima-java), an implementation based on algorithms from [9], which allows us to normalize any structure from full Boolean logic into conjunctive normal form (CNF, see Table 1). For the purpose of query translation, the OmniQuery Library implements a class for each query formalism, which invokes the transformation of the respective language into the OmniQuery format and vice versa. Those transformation implementations consume an interface provided by OmniQuery so that any other query language can be added to the tool belt of available translators. In the context of the GBN prototypes, we implemented two of these translator classes, *I2B2Translator* and *SamplyTranslator*. Both parse the original query formalism and build an OmniQuery object, which can then be translated into the other syntax by using the other translator. In this process, the translators address the differences that were described above, except for the CNF conversion (which is a feature of the core OmniQuery library).

4. Discussion

The idea to mediate between different data sources is not new, in particular on the "lower" level of databases. One approach is to combine heterogeneous data sources with a virtualization layer, as it has been done e.g. in SALUS [10]. However, attempts to integrate data sources on the "higher" level of heterogeneous software platforms are rare, at least to our knowledge. The EHR4CR system is capable of executing feasibility queries directly on the i2b2 database [11]. It does not, however, utilize the native i2b2 software stack to run these queries. In contrast, and to the best of our knowledge, our approach is the first successful attempt of interconnecting heterogeneous cohort selection platforms directly by utilizing automatic query translation and adhering to the platforms' native interfaces.

The intention behind OmniQuery was to create a generic method to translate queries between arbitrary cohort selection platforms and allow usage of different DWHs. Thus far, the library has only been used in the two GBN prototypes to translate between i2b2 and Samply. While most cohort selection platforms build on a set of features that is very similar to those two systems, OmniQuery is not fully generic. For instance, it is lacking support for advanced features that are not present in one of the systems (such as temporal constraints). In the case of our two prototypes this has not been a problem as it successfully translated queries that met the requirements of the GBN demonstrator in terms of query complexity. Both prototypes were demonstrated

to the GBN consortium in February 2017, and they were able to translate and execute queries. The system currently supports querying categorical data elements, such as "gender" or "type of sample", as well as numeric data elements, which is in particular useful for lab values.

The follow-up project German Biobank Alliance will dictate future directions of OmniQuery. However, ongoing efforts in BBMRI are indicating that its platform will also be based on Samply for injecting and distributing queries in the European network. In contrast to this pilot, it is currently planned to use MOLGENIS [12] for local data integration and as a local DWH. We plan to investigate whether OmniQuery could also act as a bridge between Samply and MOLGENIS.

5. Software Availability

The source code of OmniQuery is available on GitHub (https://github.com/German-Biobank-Node/OmniQuery).

6. Acknowledgements

The present work has been funded by the Federal Ministry of Education and Research of Germany within the project *German Biobank Node* (project number 01EY1301). It was performed in (partial) fulfillment of the requirements for obtaining the degree "Dr. rer. biol. hum." from the Friedrich-Alexander-Universität Erlangen-Nürnberg (SM).

References

[1] Xu J, Rasmussen LV, Shaw PL, et al., Review and Evaluation of Electronic Health Records-Driven Phenotype Algorithm Authoring Tools for Clinical and Translational Research, *J Am Med Inform Assoc* 2015, ocv070.

[2] Lablans M, Kadioglu D, Mate S, et al., Strategies for Biobank Networks, *Bundesgesundheitsblatt* **59.3** (2016), 373-378.

[3] van Ommen G-JB, Törnwall O, Bréchot C, et al., BBMRI-ERIC as a Resource for Pharmaceutical and Life Science Industries: The Development of Biobank-Based Expert Centres, *Eur J Med Genet* **23.7** (2015), 893-900.

[4] Majeed RW, Röhrig R., Automated Realtime Data Import for the i2b2 Clinical Data Warehouse: Introducing the HL7 ETL Cell, *Stud Health Technol Inform* **180** (2012), 270-4.

[5] Ganslandt T, Mate S, Helbing K, et al., Unlocking Data for Clinical Research – The German i2b2 Experience, *Appl Clin Inform* **2.1** (2011), 116.

[6] Lablans M, Kadioglu D, Muscholl M, et al., Exploiting Distributed, Heterogeneous and Sensitive Data Stocks While Maintaining the Owner's Data Sovereignty, *Methods Inf Med* **54.4** (2015), 346-352.

[7] Ahlbrandt J, Brammen D, Majeed RW, et al., Balancing the Need for Big Data and Patient Data Privacy - An IT Infrastructure for a Decentralized Emergency Care Research Database, *Stud Health Technol Inform* **205** (2013), 750-754.

[8] Kadioglu D, Weingardt P, Ückert F, et al., Samply.MDR – Ein Open-Source-Metadaten-Repository, *German Medical Science GMS Publishing House* 2016, doi:10.3205/16gmds149.

[9] Russell SJ, Norvig P, Artificial Intelligence, Prentice Hall, 2010.

[10] Sun H, Depraetere K, De Roo J, et al., Semantic Processing of EHR Data for Clinical Research, *J Biomed Inform* **58** (2015), 247-259.

[11] Bache R, Miles S, Taweel A, An Adaptable Architecture for Patient Cohort Identification From Diverse Data Sources, *J Am Med Inform Assoc* **20.e2** (2013), e327-e333.

[12] Swertz MA, Dijkstra M, Adamusiak T, et al., The MOLGENIS Toolkit: Rapid Prototyping of Biosoftware at the Push of a Button. *BMC Bioinformatics* **11.12** (2010), S12.