German Medical Data Sciences: Bringing Data to Life
R. Röhrig et al. (Eds.)

doi:10.3233/SHTI210053

HIStream-Import: A Generic ETL Framework for Processing Arbitrary Patient Data Collections or Hospital Information Systems into HL7 FHIR Bundles

Raphael W. MAJEED[1,a,b] , Mark R. STÖHR [a]
and Andreas GÜNTHER [a]
[a] Universities Gießen and Marburg Lung Center (UGMLC), Giessen, Germany
[b] Institute of Medical Informatics, Medical Faculty of RWTH University Aachen, Aachen, Germany

Abstract. Data integration is a necessary and important step to perform translational research and improve the sample size beyond single data collections. For health information, the most recent established communication standards is HL7 FHIR. To bridge the concepts of "minimal invasive" data integration and open standards, we propose a generic ETL framework to process arbitrary patient related data collections into HL7 FHIR – which in turn can then be used for loading into target data warehouses. The proposed algorithm is able to read any relational delimited text exports and produce a standard HL7 FHIR bundle collection. We evaluated an implementation of the algorithm using different lung research registries and used the resulting FHIR resources to fill our i2b2 based data warehouse as well an OMOP common data model repository.

Keywords. Data integration, standardization, factual databases.

1. Introduction

For research networks, data integration is a necessary and important step to perform translational research and improve the sample size beyond single data collections. Similar to other research networks, the German Centre for Lung Research (DZL) has numerous heterogeneous data collections which are individually managed by their respective owners. Aim of the DZL's data integration effort is to provide a single central data warehouse frontend, where all patient related data are combined and can be accessed by any researcher in the network.

We refer to data integration as "the computational solution allowing users, from end user (GUI) to power users (API), to fetch data from different sources, combine, manipulate and re-analyze them as well as being able to create new datasets and share these again with the scientific community." [1]. Data discovery as the first step of data integration and issue of finding the data relevant to a project [2] was covered in a previous survey within our research network [3]. The subsequent processes for data

[1] Corresponding Author, R.W.Majeed, E-mail: Raphael.Majeed@chiru.med.uni-giessen.de.

integration are applied between “operational source systems and the data presentation area” and “are commonly known as extract-transform-load (ETL)” [4,5].

As users and clinicians can be alienated by this complex technical process, we desire a “minimal invasive” approach to data integration in the sense that no modifications are required to the original data collections or corresponding software. Simultaneously, the FAIR guiding principles recommend using open standards, protocols and vocabularies for scientific data [6]. For health information, the most recent established communication standards is HL7 FHIR [7,8]. To bridge the concepts of “minimal invasive” data integration and open standards, we propose a generic ETL framework to process arbitrary patient related data collections into HL7 FHIR – which in turn can then be used for loading into target data warehouses.

2. Method

During the data discovery process, we identified a total of 68 relevant patient related data collections in our research network [3]. These data collections used heterogeneous software, yet all of which supported full data exports into (multiple) delimited text files (e.g. comma separated values CSV, tab separated values). Thus, the input format for our data integration process was settled to unmodified delimited text files.

Endpoint for data integration is most commonly a data warehouse or data repository. As different solutions exist for biomedical data warehouses (e.g. i2b2, tranSMART, OHDSI OMOP CDM), we settled on HL7 FHIR as a common intermediate output format. More specifically, a “Bundle collection” is used within the FHIR standard to communicate different entities (e.g. patient, encounter, observation) in a single resource.

With delimited text files as common input format and HL7 FHIR Bundle resource as standard output format, the following steps are needed: (a) Identify and abstract relations of patient data collections in order to produce a generic model for corresponding data exports; (b) design algorithm to efficiently parse abovementioned relations and (c) evaluate feasibility of algorithm with real data.

3. Results

3.1. Abstract relational model

Analysis of export formats of aforementioned data collections resulted in three common relational tables: (i) general patient information like e.g. patient identifier, birth date, age, gender. (ii) encounter or visit related information like e.g. encounter identifier, patient identifier, date and time of encounter, type of encounter (in-/outpatients, emergency). (iii) individual data points / observations / measurements / surveys commonly with timestamp, value, unit. In some cases, a data point consists of multiple parts like e.g. two values for a blood pressure measurement or in case of medication: dosage, route, substance. The abstract relational model between these tables is shown in figure 1.

Patient tables commonly contain one row per patient. Encounter tables respectively one row per encounter or visit. The fact tables are further classified into “wide tables” and “long tables”. On the one hand, in wide tables each row contains multiple information points like e.g. in a survey one row per survey with one column per question/answer. On the other hand, “long tables” contain few columns and multiple rows per entity, like e.g. for a survey one row for each question/answer.

With this abstract relational model, we were able to describe exports for all examined data collections.

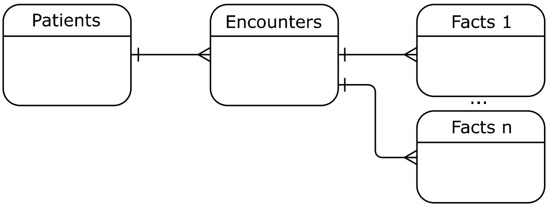

Figure 1. Abstract relational model for patient related data export formats. For one patient, there can be multiple encounters. For each encounter, there can be multiple facts in different fact tables.

3.2. Generic algorithm for parsing table based exports

To process arbitrary patient related data tables into a common target format, each row must be read at least once. In an optimal algorithm, each row is read exactly once. The following algorithm makes three assumptions: (a) patient and encounter tables contain identifiers which are referenced from the fact tables. (b) in all tables same patient ids and same encounter ids are grouped together. (c) patient and encounter identifiers follow a consistent order in all table files.

With these assumptions, the algorithm works as follows: (1) load first/next patient row, (2) output FHIR patient resource, (3) load first/next encounter row, (4) output FHIR encounter resource, (5) if encounter row has different patient id than patient resource, go to (1) otherwise continue to (6) load first/next fact row from fact table 1. If fact row has different encounter id than encounter resource, go to next fact table. Otherwise output FHIR observation resource and continue with (6) from same fact table. For each fact table repeat from (6) until last fact table, then go to (3) until no more encounter rows are available. The algorithm diagram is depicted in figure 2.

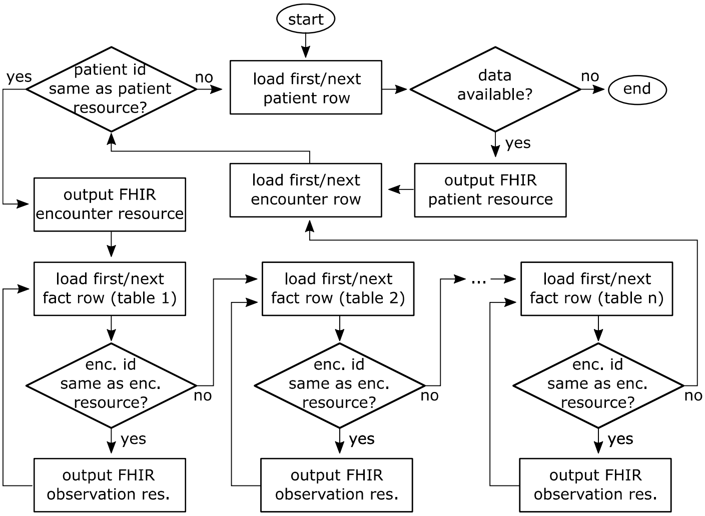
start
load first/next patient row
data available?
no
end
yes
output FHIR patient resource
load first/next encounter row
patient id same as patient resource?
no
yes
output FHIR encounter resource
load first/next fact row (table 1)
enc. id same as enc. resource?
no
yes
output FHIR observation res.
load first/next fact row (table 2)
enc. id same as enc. resource?
no
yes
output FHIR observation res.
...
load first/next fact row (table n)
enc. id same as enc. resource?
no
yes
output FHIR observation res.

Figure 2. Algorithm flow chart.

During the algorithm, loaded rows which do not match the current patient/encounter resource are kept until the next (matching) patient/encounter is loaded in order to prevent unnecessary row skips. This algorithm processes each row from each file exactly once.

For data exports which do not meet the above-mentioned assumptions (a)-(c), the individual export files can be sorted using standard sorting tools and therefore be converted to meet the assumptions.

3.3. Algorithm feasibility

We implemented this algorithm in a Java application. For input, the algorithm needs file names for patient table file, encounter table file and each fact table file as well as column names for patient identifier, encounter identifier, timestamps and values. This information is provided in XML format to the Java implementation.

Using our previously identified data collections [3], we validated the algorithm multiple text data export variants. To satisfy the algorithm's preconditions, data preprocessing in the form of sorting and filtering duplicates was required in many cases. In all cases, the algorithm performed as expected and produced valid FHIR resource bundles.

4. Discussion

With the provided algorithm, relational data exports with multiple text files can be converted to standard FHIR bundle collection resources.

We used this algorithm in the German Centre for Lung Research (DZL) to process data sources from Excel, Access to File Maker, SecuTrial, RedCap and proprietary SQL databases. As mentioned in the results, preprocessing is needed in many cases to sort and group rows accordingly. In all cases, this preprocessing was accomplished using standard command line tools or in some cases export configurations from the original software. In two cases, we processed XML exports using XSLT to generate row based text files. With this method, the scope of the algorithm can be extended also to XML based exports.

Data integration of heterogeneous data collections and conversion to FHIR resources can also be accomplished using existing ETL software tools like Talend Open Studio or Pentaho. Yet, these tools not specific to the medical field and require informatics experts with programming skills. On the other hand, our algorithm is specifically tailored to patient related data and can be run by medical documentation officers without programming experience.

Once data collections are converted to standard HL7 FHIR resources, additional tools can be used for interaction and integration with data warehouses common in the biomedical field. For example, solutions exist to import FHIR resources into i2b2 data warehouse [9]. By choosing FHIR resources as intermediate format, the data can also be loaded into FHIR servers for sharing and further processing.

In our case, we transferred the resulting data into an i2b2 data warehouse. For a different registry, we used our algorithm to transfer data collections into an OMOP common data model [10].

In summary, the presented algorithm provides a feasible and generic way to process heterogeneous data collections via text based exports into a common standard representation for health information interchange.

References

[1] V. Lapatas, M. Stefanidakis, R.C. Jimenez, A. Via, M.V. Schneider. Data integration in biological research: an overview. J Biol Res (Thessalon) 2015. doi:10.1186/s40709-015-0032-5.

[2] J. Hendler. Data Integration for Heterogenous Datasets. Big Data. 2014;2:205–15.

[3] R.W. Majeed, M.R. Stöhr, C. Ruppert, A. Günther. Data Discovery for Integration of Heterogeneous Medical Datasets in the German Center for Lung Research (DZL). Stud Health Technol Inform. 2018;253:65-69.

[4] R. Kimball. The data warehouse toolkit. New York: John Wiley & Sons; 1996.

[5] C. Weng, C. Clinical data quality: a data life cycle perspective. Biostatistics & Epidemiology. 2020; 4(1), 6-14.

[6] M.D. Wilkinson, M. Dumontier, I.J. Aalbersberg, G. Appleton, M. Axton, A. Baak, et al. The FAIR Guiding Principles for scientific data management and stewardship. Scientific Data. 2016;3:160018 EP.

[7] D. Bender, K. Sartipi. HL7 FHIR: An Agile and RESTful approach to healthcare information exchange. Proceedings of the 26th IEEE international symposium on computer-based medical systems (2013) 326-331.

[8] M.L. Braunstein. FHIR. In: Health Informatics on FHIR: How HL7's New API is Transforming Healthcare. Springer, Cham, 2018. S. 179-203.

[9] A. Boussadi, E. Zapletal. A fast healthcare interoperability resources (FHIR) layer implemented over i2b2. BMC medical informatics and decision making 17.1 (2017): 120.

[10] P. Fischer, M.R. Stöhr, H. Gall, A. Michel-Backofen, R.W. Majeed. Data integration into OMOP CDM for heterogeneous clinical data collections via HL7 FHIR Bundles and XSLT. Stud Health Technol Inform. 2020