Challenges of Trustable AI and Added-Value on Health
B. Séroussi et al. (Eds.)

doi:10.3233/SHTI220476

FAIRifying a Quality Registry Using OMOP CDM: Challenges and Solutions

Daniel PUTTMANN[a,b,1], Nicolette DE KEIZER[a,b], Ronald CORNET[b], Eric VAN DER ZWAN[a,b] and Ferishta BAKHSHI-RAIEZ[a,b]
[a] *The National Intensive Care Evaluation (NICE) registry, The Netherlands*
[b] *Department of Medical Informatics, Amsterdam Public Health Research Institute, Amsterdam UMC, University of Amsterdam, The Netherlands*

Abstract. The need for health data to be internationally Findable, Accessible, Interoperable and Reusable (FAIR) and thereby support integrative analysis with other datasets has become crystal clear in the ongoing pandemic. The Dutch National Intensive Care Evaluation (NICE) quality registry adopted the Observational Medical Outcomes Partnership Common Database Model (OMOP CDM) to achieve a FAIR database. In the process of adopting the OMOP CDM, many modeling, technical, and communication challenges needed to be solved. Through communication with the OMOP CDM implementation community, previously done research and trial-and-error we found solutions that we believe can help other healthcare institutions, especially ICU quality registries, FAIRify their databases.

Keywords. OMOP CDM, ETL Process, Quality Registry, OHDSI

1. Introduction

Though cooperation has always been important in research, the COVID-19 pandemic showed just how important it is to federate local databases for large-scale analysis efficiently. Both for research and healthcare, large amounts of data were and are still needed to combat outbreaks, to develop novel treatments and to gain insight into the new disease. However, time is short, and data is fragmented.

In general, health data is recorded in hospitals in a flexible, unstructured format or in a proprietary structured format. Databases are modeled with a clear, locally implemented goal in mind. This makes the data convenient for its original use case, but difficult to reuse for other purposes [1]. The databases of healthcare institutions are often modeled in this way, making them poorly interoperable.

Since the '90s, researchers and standard development organizations have developed models to standardize healthcare databases. As a result, communities have grown from the different standardization paradigms, one being the Observational Health Data Sciences and Informatics (OHDSI) program [2]. The OHDSI program is an international collaborative aiming to analyze and perform research on large amounts of health data. It developed the Observational Medical Outcomes Partnership Common Database Model (OMOP CDM) to facilitate interoperability between databases. Interoperability here

[1] Corresponding Author, Daniel Puttmann; E-mail: d.p.puttmann@amsterdamumc.nl.

means that databases can be used to collaborate and combine data by using the same format, with enough metadata to make unfamiliar data sources findable and reusable. OMOP CDM can be used to work towards a Findable, Accessible, Interoperable and Reusable (FAIR) data infrastructure [3]. By making health data FAIR, medical research could make better use of technical advancements such as machine learning and the data sharing across the internet. In Europe, the European Health Data & Evidence Network (EHDEN) was launched to facilitate international, large-scale research with real-world health data. They have been collaborating with OHDSI since their inception in 2018.

The National Intensive Care Evaluation (NICE) registry is a quality registry developed by Dutch ICUs to monitor and improve the quality of care by learning from other ICUs. Despite some cross-registry projects, international collaboration with other ICU registries such as the Australian ANZICS-CORE or the Sri Lankan Critical Care Asia (CCA) is hampered by use of local definitions and lack of interoperability [4].

By adopting OMOP CDM and by joining the EHDEN-OHDSI community, NICE aimed to make its data FAIR and accessible for international researchers and align itself with other ICU registries adapting the OMOP, such as the CCA. With this, we let the ICU community benefit from larger datasets.

The aim of this paper is firstly to describe the different steps of the Extract, Transform and Load (ETL) process to make the NICE database FAIR by using OMOP CDM and OHDSI tools and secondly to investigate the challenges that were faced and the solutions that were found during the ETL process.

2. Methods

We performed the ETL on six tables from the NICE database [5]. In total there were 158 data elements adapted in English to the OMOP CDM. The FAIRification process was divided into four steps (see Figure 1). To complete the various steps, we used the Book of OHDSI, EHDEN academy's ETL course, OHDSI's GitHub wiki, and the OHDSI tools [6-9].

In the first step, we used the White Rabbit, Rabbit-In-A-Hat and Usagi OHDSI tools to analyze and map the data to the OMOP CDM and its underlying standard vocabularies.

The second step was the Extract, Transform and Load (ETL) process, which we performed using twelve Microsoft SQL Server (MSSQL) scripts. The database needed to be configured with empty tables representing the OMOP CDM and constraints were set for primary and foreign key rules. The OMOP CDM tables were populated based on the results from step one.

In the third step, we validated the ETL by performing two quality checks, for each of OHDSI’s data quality management tools. These were the Data Quality Dashboard and the ACHILLES Heel. We evaluated the error messages and deleted the offending rows from the database where appropriate, per OHDSI’s recommendation. Lastly, we implemented the Atlas research platform to allow for data analysis on our OMOP CDM database and we published a link to our database on the EHDEN data portal for researchers to contact us from.

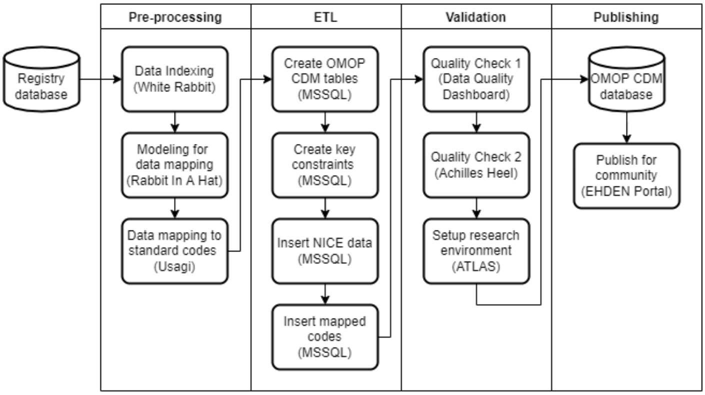
Pre-processing
ETL
Validation
Publishing
Registry database
Data Indexing (White Rabbit)
Modeling for data mapping (Rabbit In A Hat)
Data mapping to standard codes (Usagi)
Create OMOP CDM tables (MSSQL)
Create key constraints (MSSQL)
Insert NICE data (MSSQL)
Insert mapped codes (MSSQL)
Quality Check 1 (Data Quality Dashboard)
Quality Check 2 (Achilles Heel)
Setup research environment (ATLAS)
OMOP CDM database
Publish for community (EHDEN Portal)

Figure 1. A flowchart of the process from the registry database to the OMOP CDM database. The tools used are mentioned in parentheses. All tools except MSSQL are developed by OHDSI.

3. Results

In this section we describe the challenges and the solutions, categorized in five main topics, namely mapping problems, differences in unit of observation, differences in structure of the database, security concerns and technical restrictions.

Mapping problems: we were unable to map 50 out of 158 data elements because no standardized code was available to fully reflect the concept's meaning. Five of these data elements and their values were part of specialized code sets such as for ICUs commonly used APACHE IV reasons for admission classification [10]. Finding a solution to this was challenging due to the lack of documentation on what to do when no suitable code was available. On OHDSI forums we found two solutions based on a conference poster [11]:

- Use a lookup table to insert custom codes in available tables after inserting source codes or headers
- Insert custom codes at the end of the imported OHDSI standard codes and create relationships such as 'maps to' in the concept relationship table

We implemented a hybrid form where each previously unmapped data element and value of the specialized code sets were assigned custom codes and code relationships. They were then inserted in the right OMOP CDM tables using a lookup table.

Unit of observation: the NICE database was centered around ICU visits, not around patients. This was an issue when generating primary keys for OMOP CDM tables, which made linking the OMOP CDM tables difficult. This was solved by generating unique keys for each part of the NICE database's composite primary key, used to designate unique ICU visits. This key included an encrypted patient id, a hospital admission number, and an ICU admission number. Respectively, for the person, visit occurrence and visit detail tables we created three lookup tables that generated primary keys from different combinations of the NICE primary keys and some patient characteristics if necessary.

Structure of the database: the NICE tables were in a wide format while the OMOP CDM tables were in a long format. Every data item name was a header in the NICE table, while the names were put in a single column in the OMOP CDM. This was eventually solved by creating a temporary table with the desired data elements before each insertion and using the 'UNPIVOT' SQL operator. It takes multiple columns and collates them into two columns: one column for the header and another for the value belonging to that header. If the data type of the source value did not match with the data type of the target column, we converted the source value to that data type. We then inserted the data into the target table using the unpivoted temporary table.

Security concerns and technical restrictions: the use of the university infrastructure with all its constraints lead to delays in the ETL process. The data quality tools validating the database required R scripts to run, which required specific versions of certain R and java packages to be installed. These installations needed to be approved by the IT department due to protocol. Furthermore, we chose to implement ATLAS via a Docker container because of the reported difficulty in implementation. The ATLAS Docker container is a package containing ATLAS and all its dependencies. The Docker also included a major dependency called WebAPI that connects the OMOP CDM database server to its own server. We were reluctant to install the WebAPI on the NICE database as it was not intended to be connected to the internet. These concerns were a result of a misnomer. The term WebAPI usually refers to a way for external applications to interface with a data source. Although the WebAPI java application has the possibility to interface with such applications, it can also be installed and used locally. To run ATLAS, 104 new tables were needed to store results in. There was no script on the OHDSI Github that could generate these tables, but we could acquire a script from the OHDSI community.

The last *technical hurdle* was that, when creating a cohort to do analysis on, we were not able to use the function to create inclusion criteria due to key duplication errors. After consulting the OHDSI implementation forums we found a solution that had us delete primary keys in the ATLAS cohort inclusion tables [12]. The issue was that these tables wrongly had primary keys assigned to them.

4. Discussion and Conclusions

By implementing OMOP CDM, NICE can now use the powerful ATLAS and R toolkit of OHDSI to do research and share their data with researchers across the globe. Our future research should focus on the capabilities of the new database and the further standardization of the data elements.

OHDSI's ETL process had from an implementer's perspective a very low technical cohesion. Meaning that in each step new applications had to be installed, new programming languages had to be used and new documentation had to be searched for on different platforms. We suggest OHDSI to focus on the coherence of their ETL process to improve the rate of adaptation. More healthcare institutions will be able to adopt OMOP CDM if the materials and documentation are not as fragmented as they are now, and if software works without having to install many specific dependencies or writing additional code.

To solve documentation fragmentation, the EHDEN-OHDSI collaboration developed the Book of OHDSI and EHDEN academy [6,7]. However, they conflate theory and practice. Ideally, the Book of OHDSI would be a main source for background information and the EHDEN academy a source for implementation guides. Moreover, OHDSI's tutorials can be followed with an Amazon virtual machine called OHDSI-in-a-box, which simulates a database and all ETL tooling to lead users through the process using fake data. However, to our knowledge there are no alternatives for implementers without an Amazon paid subscription.

Technical difficulties could be solved with the creation of Docker containers for every application. The OHDSI community has already started a project on this called BROADSEA [13]. If OHDSI's Docker containers worked out of the box, it would not only accelerate the process, but also make the task less daunting.

We expect this paper to be of use to most other health organizations that want to FAIRify their database using the OMOP CDM. The challenges we faced had very common causes: many medical databases have a different unit of observation, a different structure, or data elements with specific, hard to standardize meanings. When mapping data, the list of data elements should be inspected for unmappable codes. These should be set aside and mapped to custom codes in the format of the OMOP CDM concept table. During the ETL, implementers should have a test OMOP CDM database and fill it with a month's data from the source database. It is also important to identify the transformations that the source data needs to undergo during the ETL, such as an unpivot.

Lastly, we recommend using the Docker containers for the validation step if they are available. An alternative solution is to prepare a list of dependencies with their version before trying to run the application. Throughout the entire process, data security should be closely monitored. Test data should be carefully generated and anonymized. It should also be clear what access should be given for each application since some need more than others. If other health organizations, especially ICU quality registries, want to join the NICE in joining the European science community, we believe that our experiences can help with their transition.

References

[1] van der Lei J. Use and abuse of computer-stored medical records. Methods of information in medicine. 1991;30(02):79-80.

[2] OHDSI – Observational Health Data Sciences and Informatics [Internet]. Ohdsi.org. 2022. Available from: https://www.ohdsi.org/

[3] de Groot R, Cornet R, de Keizer N, Benis N, Raiez F. OMOP CDM compared to ContSys (ISO13940) to make data FAIR – OHDSI [Internet]. Ohdsi.org. 2022. Available from: https://www.ohdsi.org/2020-global-symposium-showcase-52/

[4] Dongelmans D, Pilcher D, Beane A, Soares M, del Pilar Arias Lopez M, Fernandez A et al. Linking of global intensive care (LOGIC): An international benchmarking in critical care initiative. Journal of Critical Care. 2020;60:305-310.

[5] NICE Data Dictionary [Internet]. NICE. 2022. Available from: https://www.stichting-nice.nl/dd/#start

[6] Informatics O. The Book of OHDSI [Internet]. Ohdsi.github.io. 2022. Available from: https://ohdsi.github.io/TheBookOfOhdsi/

[7] EHDEN Academy [Internet]. Academy.ehden.eu. 2022. Available from: https://academy.ehden.eu/

[8] OHDSI Common Data Model Wiki [Internet]. Ohdsi.github.io. 2022. Available from: https://ohdsi.github.io/CommonDataModel/index.html

[9] Observational Health Data Sciences and Informatics GitHub [Internet]. GitHub. 2022. Available from: https://github.com/OHDSI

[10] Zimmerman JE, Kramer AA, McNair DS, Malila FM. Acute Physiology and Chronic Health Evaluation (APACHE) IV: hospital mortality assessment for today's critically ill patients. Critical care medicine. 2006 May 1;34(5):1297-310.

[11] Philofsky M, RN, MS Mapping Custom Source Codes to Standard Concepts: A Comparison of Two Approaches [Poster]. 2020 Available from: https://www.ohdsi.org/2020-global-symposium-showcase-18/

[12] Tystan, Knoll C, Brandt P. Cohort Generation in Atlas with Generation status Failed [Internet]. OHDSI Forums. 2020. Available from: https://forums.ohdsi.org/t/cohort-generation-in-atlas- with-generation-status-failed/5690/31

[13] Evans L, Suchard M. OHDSI Broadsea, GitHub Repository [Internet]. 2021. Available from: https://github.com/OHDSI/Broadsea