Caring is Sharing – Exploiting the Value in Data for Health and Innovation
M. Hägglund et al. (Eds.)

doi:10.3233/SHTI230061

Breaking Barriers for Interoperability: A Reference Implementation of CSV-FHIR Transformation Using Open-Source Tools

Samer ALKARKOUKLY[a,b,1], Md. Mostafa KAMAL[a,b] and Oya BEYAN[a,c]
[a] *Institute for Biomedical Informatics, University of Cologne, Faculty of Medicine and University Hospital Cologne, Cologne, Germany*
[b] *CECAD, University of Cologne and University Hospital Cologne, Cologne, Germany*
[c] *Department of Data Science and Artificial Intelligence, Fraunhofer FIT, Germany*
ORCiD ID: Samer Alkarkoukly https://orcid.org/0000-0003-1891-1366, Md. Mostafa Kamal https://orcid.org/0000-0002-5506-4875, Oya Beyan https://orcid.org/0000-0001-7611-3501

Abstract. FHIR is a widely accepted interoperability standard for exchanging medical data, but data transformation from the primary health information systems into FHIR is usually challenging and requires advanced technical skills and infrastructure. There is a critical need for low-cost solutions, and using Mirth Connect as an open-source tool provides this opportunity. We developed a reference implementation to transform data from CSV (the most common data format) into FHIR resources using Mirth Connect without any advanced technical resources or programming skills. This reference implementation is tested successfully for both quality and performance, and it enables reproducing and improving the implemented approach by healthcare providers to transform raw data into FHIR resources. For ensuring replicability, the used channel, mapping, and templates are available publicly on GitHub[2].

Keywords. FHIR, Fast Health Interoperability Resources, Interoperability, Data Transformation, CSV, Mirth Connect, open-source

1. Introduction

Transforming raw data from healthcare information systems into FHIR resources usually requires advanced technical skills and software. The aim of this work is to propose a user-friendly method to transform medical data from CSV format into FHIR resources without requiring sophisticated programming scripts or tools. It uses an open-source tool and publicly available transformation "channels" and templates, and could be reproduced and improved by any healthcare provider with basic knowledge and skills.

Healthcare providers collect different types of patient data and store them in Electronic Medical Records "EMRs" which are defined as computerised medical information systems that collect, store, and display patient information [1]. EMRs in healthcare institutions reflect the differences among the several information systems used

[1] Corresponding Author: Samer Alkarkoukly, E-mail: Mohammad.abboud-alkarkoukly@uk-koeln.de
[2] https://github.com/alkarkoukly/CSV-FHIR-Transformer

in the different departments of each healthcare facility, which lead the medical data to be fragmented and spread over different systems. Data fragmentation is one of the challenges against the interoperability of EMRs. To enable the secondary usage of clinical data, several health information systems providers permit data export in Comma Separated Values (CSV) format, which is the most frequently used data format in storing tabular data as a two-dimensional array. However, CSV does not describe metadata such as datatype, data quality, or data provenance [2], and CSV files fail to provide semantic interoperability with other systems or other potential users of the data.

In order to facilitate medical data exchange and to enhance interoperability, Fast Health Interoperability Resources (FHIR) were designed and used. FHIR is a standard for exchanging healthcare information electronically [3] and it maintains syntactic interoperability and ensures correct interpretation of the data at the receiver end. Also, it maintains semantic interoperability by using terminologies and coding systems. Additionally, transferring data in FHIR format enhances the FAIR principles (Findable, Accessible, Interoperable, Reusable) which guide scientific data management and stewardship, and are relevant to all stakeholders in the digital health ecosystem [4].

To enhance the reusability and interoperability of clinical data, several tools were designed to receive, filter, clean, transform, export and validate the collected data in the healthcare facilities' primary information systems. One of these open-source tools is Mirth Connect, developed in 2006 as MIRTH, an acronym for "Messaging, Interfaces, Routing, Transformations, Healthcare" [5]. It is an integration engine that reads patient data from different sources (database, local directory, local server and many others), transforms the data and translates messages standards into different ones. Mirth Connect can read and export files from different formats (CSV, JSON, XML, HL7 V2 messages).

The literature review showed that the reference implementation of this work has not been implemented yet and would be an added knowledge to the work in this field. The most relevant published work included converting HL7 ADT messages into FHIR using an open-source tool called XSLT [6], but the approach seems to be focused on oncology data and only on ADT messages. Another article focused on the concept and the logical mapping and the architectural differences between openEHR and FHIR and not detailing the technical steps followed during the transformation [7]. A third approach [8] discussed the concept and challenges against converting data to FHIR using the tool "CAMP FHIR" which can get the data from a relational database and convert to FHIR with no details about the followed steps.

Currently, most of the available applications for transforming CSV data into FHIR require a Java-based integration service, which requires advanced technical skills in Java development. In this work, the transformation was done without coding, in a user-friendly approach and interface. The additional added value of this work, is the usage of the open-source integration engine "Mirth Connect", which is widely used to manage HL7 V2 messages in healthcare facilities, in transforming data into FHIR.

2. Methods

This reference implementation uses Mirth Connect, as an open-source data integration engine, to transform CSV files exported from the healthcare information system, or filled manually during data collection, and saved in a local folder, then exports the data as FHIR Patient resources in so-called "transaction bundle" to two destinations: 1) a FHIR server. 2) a local folder in the same PC as JSON files. Figure 1 illustrates the process:

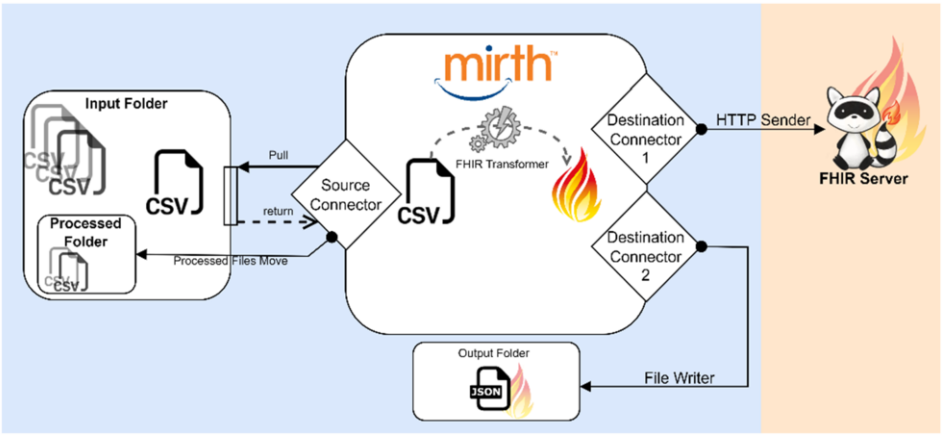

Figure 1. Illustration of Mirth Connect Channel for CSV-FHIR Transformation

Mirth Connect (MC) could be downloaded and installed from GitHub or from the vendor. The installation includes four components MC Server, MC Server Manager, MC Command Line Interface, and MC Administrator Launcher, and it will obtain two ports from the local PC. After installation, all the following work can be done using the graphical user interface of Mirth Connect by running the server manager first, starting MC service, and then launching MC administrator. A channel needs to be designed and configured in MC Administrator; the configuration settings are provided in Table 1.

Table 1. Mirth Connect CSV-FHIR Transformer Channel configuration

Connector	Data Type	Connector Type	Configuration
Source	Delimited Text	File Reader	Directory: Local PC input folder Filename Filter Pattern: "*.csv"
Destination 1	JSON	HTTP Sender	URL: FHIR server endpoint Content-type: "application/JSON" Content: "${JsonUtil.prettyPrint(${message.encodedData})}"
Destination 2	JSON	File Writer	Directory: Local PC output folder File Name: "${originalFilename}${COUNT}.json" Template: Same value of "content" of destination 1

Transformer Configuration: The source connector has a transformer which transforms the CSV inbound message from the source into JSON outbound messages as FHIR resources in a transaction bundle for both of the planned destinations. The inbound message template is a predefined CSV file reflecting the FHIR Patient resource elements and all the incoming messages should use the same template. The template is available publicly in the project's GitHub repository and could be easily extended and improved. The outbound message has a JSON structure obtained from FHIR Patient resource. The mapping between the elements of inbound CSV and outbound JSON is done easily by drag-and-drop from the inbound message tree into the outbound message tree.

After deploying the channel, Mirth Connect automatically reads the entire content of all CSV files in the input folder, transforms the content into FHIR resources, then sends the generated resources into a locally installed FHIR server and at the same time, exports them as JSON files in the output folder. All the processed CSV files will be moved into another folder "Processed". The processes of this reference implementation are just an example of the implementation of CSV-FHIR transformation, and they are illustrated in Figure 1. This scenario proves that the followed methodology could be implemented in any other use-case of raw data transformation to FHIR. The input data

could be any of the different file formats (XML, JSON, HL7, TSV, DICOM) or any of the different types of source connectors (HTTP Listener, Channel Reader, Database Reader, TCP Listener) or even a response from REST interface (such as another FHIR server). To test the channel, 21 synthetic datasets of randomised real personal data were generated in CSV files and used to test the quality and performance of the channel.

3. Results

The suggested reference implementation was tested successfully by exporting FHIR resources to a locally installed HAPI server and to a local PC directory, and additional quality and performance tests were implemented to check the applicability on large-scale implementations and also the validity and correctness of the generated FHIR resources. The performance test included datasets in CSV format from the input folder in 3 scenarios (10 files of 1,000, 10 files of 10,000, 1 file of 50,000 records); the processing speed average was 59 milliseconds per CSV record, and it is correlated with records amount. The quality test included monitoring the errors received from the channel logs, which was limited to only one timeout error during processing the largest CSV file. Additionally, a validation test of the generated FHIR resources was implemented by validating a random sample of 1% of the processed records (700 resources) using the "$validate" function of the destination server, and 0.1% (70 resources) were checked for correctness by comparing the generated resources with the original data. All the tested resources were valid with no warnings or errors, and matched the original data.

During the implementation, none of the programming languages was needed neither in installation (done with the graphical user interface) nor in the channel set-up (done by drag and drop between the inbound and outbound elements). All the used files, including the exported channel, CSV and JSON templates, and the testing CSV files, are available publicly on GitHub https://github.com/alkarkoukly/CSV-FHIR-Transformer.

4. Discussion

Using Mirth Connect as an open-source tool showed full success in transforming patient data from CSV format into FHIR Patient resources represented in transaction bundles. This approach is suitable for all types of healthcare facilities planning to generate FHIR resources from their primary information systems using basic technical skills and IT infrastructure, without the need for advanced hardware or virtual machines, and without any external services built with Java, C, or Python. The approach is applicable on different types of data sources (regular export in CSV, SQL queries, live streams of files or messages, and services with REST APIs). The reference implementation targeted Patient resource from FHIR specifications, and it is applicable on all of the other types of FHIR resources. It is also possible to include data cleaning, files filtering, data values mapping to terminologies or value sets, and variable format transformations in the channel of Mirth Connect. This could be done using the JavaScript programming environment contained in Mirth Connect user interface. Although the input CSV structure is static, it can be configured with the user interface to meet the different needs. The limitation of this approach is mainly related to the communication with the destination FHIR server and interactively exchange data in both ways during the submission of each single resource. For example, the need for checking if a specific

resource is already available in the FHIR server and then to conditionally create or update the intended resource based on specific elements. This situation is still theoretically possible with Mirth Connect by processing the response from the server. However, it requires further research and testing which are not covered in this paper. The approach followed in this work is reproducible, repeatable and improvable by using the exported channel, templates and testing files provided in the project repository in GitHub.

5. Conclusions

Transforming raw data from CSV into FHIR resources usually requires advanced technical and programming skills, and is not easily applicable alone by the healthcare provider or the data scientist. By using Mirth Connect as an open-source and user-friendly mapping and transformation tool, this reference implementation proves the possibility of transforming CSV into FHIR Patient resources and sending them to a FHIR server and a local directory. This work successfully tested the quality and performance of the implemented approach on large-scale implementations.

The channel, templates, and testing files used in this work are available publicly, and could be used to reproduce and improve the implemented approach. Future research on the topic can demonstrate further implementation possibilities with other types of FHIR resources and including advanced filtering, transformation and mapping processes.

Acknowledgment
This work was partially supported by the HiGHmed Consortium funded by the German Federal Ministry of Education and Research (BMBF, funding code [01ZZ1802U]).

References

[1] Kubben P, Dumontier M, Dekker A. Fundamentals of clinical data science. Fundamentals of Clinical Data Science. Springer International Publishing; 2018;1–219.
[2] Nayak A, Božić B, Longo L. Data Quality Assessment of Comma Separated Values Using Linked Data Approach. 2022. p. 240–50.
[3] Overview - FHIR v4.3.0 [Internet]. [cited 2022 Aug 23]. Available from: https://www.hl7.org/fhir/overview.html
[4] Martínez-García A, Cangioli G, Chronaki C, Löbe M, Beyan O, Juehne A, et al. FAIRness for FHIR: Towards Making Health Datasets FAIR Using HL7 FHIR. Stud Health Technol Inform [Internet]. Stud Health Technol Inform; 2022 [cited 2022 Aug 23];290. Available from: https://pubmed.ncbi.nlm.nih.gov/35672963/
[5] Healthcare Integration Engine | Mirth® Connect by NextGen Healthcare [Internet]. [cited 2022 Nov 28]. Available from: https://www.nextgen.com/products-and-services/integration-engine
[6] Deppenwiese N, Delpy P, Lambarki M, Lablans M. ADT2FHIR - A Tool for Converting ADT/GEKID Oncology Data to HL7 FHIR Resources. Stud Health Technol Inform [Internet]. Stud Health Technol Inform; 2021 [cited 2022 Aug 29];283:104–10. Available from: https://pubmed.ncbi.nlm.nih.gov/34545825/
[7] Fette G, Ertl M, Störk S. Translating openEHR Models to FHIR. Stud Health Technol Inform [Internet]. Stud Health Technol Inform; 2020 [cited 2022 Aug 29];270:1415–6. Available from: https://pubmed.ncbi.nlm.nih.gov/32570686/
[8] Pfaff ER, Champion J, Bradford RL, Clark M, Xu H, Fecho K, et al. Fast Healthcare Interoperability Resources (FHIR) as a Meta Model to Integrate Common Data Models: Development of a Tool and Quantitative Validation Study. JMIR Med Inform [Internet]. JMIR Med Inform; 2019 [cited 2022 Aug 29];7. Available from: https://pubmed.ncbi.nlm.nih.gov/31621639/