pHealth 2022
B. Blobel et al. (Eds.)

doi:10.3233/SHTI220974

ETL Processes for Integrating Healthcare Data - Tools and Architecture Patterns

Ka Yung CHENG [a,b,1], Santiago PAZMINO [a,b] and Björn SCHREIWEIS [a,b]
[a] *Institute for Medical Informatics and Statistics, Kiel University, Germany*
[b] *University Hospital Schleswig-Holstein, Kiel, Germany*

Abstract. Improving the interoperability of healthcare information systems is a crucial clinical care issue involving disparate but coexisting information systems. However, healthcare organizations are also facing the dilemma of choosing the right ETL tool and architecture pattern as data warehouse enterprises. This article gives an overview of current ETL tools for healthcare data integration. In addition, we demonstrate three ETL processes for clinical data integration using different ETL tools and architecture patterns, which map data from various data sources (e.g. ME-ONA and ORBIS) to diverse standards (e.g. FHIR and openEHR). Depending on the project's technical requirements, we choose our ETL tool and software architecture pattern to boost team efficiency.

Keywords. Health Information Exchange, ETL tools, Architecture patterns, HiGHmed, Software Design

Introduction

German Medical Informatics Initiative (MII) aims to achieve semantic interoperability of clinical data for reuse in patient care and biomedical research [1, 2]. For this clinical data integration and internal data retrieval, at the University Hospital Schleswig-Holstein (UKSH) Medical Data Integration Center (MeDIC) and as partner of HiGHmed Consortium [3], we require ETL (Extract-Transform-Load) processes to extract valuable clinical data and convert them into international exchange standards, e.g. Fast Healthcare Interoperability Resources (FHIR) and openEHR. Clinical data information is extracted from various data sources originating from our hospital information systems, such as exporting data from ORBIS as HL7v2 messages. Before embarking on any software development project, one of the first questions to ask relates to our choice of tools and architecture.

1. State of the Art

ETL stands for extract, transform, and load [4]. It aims for sorting, cleansing and standardizing data from multiple heterogeneous data sources to single or multiple sources. In this way, it improves productivity and data accuracy in data warehouses. In Business In-telligence (BI), ETL processes are particularly used to collect data from various

[1] Corresponding Author: Ka Yung Cheng, Institute for Medical Informatics and Statistics, University of Kiel, Germany; Email: KaYung.Cheng@uksh.de

sources and information systems and process it syntactically and semantically; thus, making it valuable for users. In recent years, studies [4–7] have analyzed ETL tools related to busi-ness intelligence. [8–12] demonstrate the development of ETL processes for healthcare data transformation. However, there is a literature gap in comparing the ETL tools and software architecture with practical healthcare examples.

Common healthcare data exchange formats are: 1) FHIR, 2) OpenEHR via 3) Integrating the Healthcare Enterprise (IHE) profile Cross-enterprise document sharing (XDS).

1) Fast Healthcare Interoperability Resources (FHIR) organizes information into discrete and independent data elements and using the REST approach, Resources (individual information packets) can be created and shared rapidly and easily.
2) openEHR [13] describes the management and storage, retrieval and exchange of health data by using archetype-based Electronic Health Records (EHRs). Archetypes provide semantic modelling by referencing particular Reference Models (RMs) defining possible arrange- ments of data that correspond to logical data points and groups for a domain topic.
3) XDS [14] exchanges any kind of document by separating document content (e.g. unstructured PDF documents) from metadata (structured and searchable object characteristics) and specifies the use of SOAP and SOAP with Attachments for communicating between actors.

2. Concept

For the conceptual design of data transformation, we show common ETL tools for health-care data. In addition, we choose three ETL processes within our MeDIC as practical examples and investigate from where our data originates, to which system/ data formats they have to be converted and published.

2.1. Common ETL Tools for Healthcare Data

Generally, ETL tools can be divided into two categories [4, 5]: *tool-based* and *source-code based*. Tool-based ETL processes facilitate creating workflows of ETL activities and automating the execution with a graphical user interface (GUI). Among them, several open-source data integration tools achieve high levels in terms of maturity and performance in healthcare use cases as follows:

- Talend [7] is an open-source tool but not a full BI suite. It uses a code-generating approach. Besides high data connectivity, it allows writing more customized SQL queries and Java from its own GUI.
- Apache NiFi supports scalable data routing and automates the data flow between systems. It has a web-based user interface. Additionally, it allows for extending the core functionality by developing customized processors. For example, data science platform [8] uses Nifi to route data flow
- Pentaho Data Integration (PDI) is an open-source application for Business Intelligence with reporting, analysis, dashboard, datamining, and ETL capabilities. Loma Linda University Health care, Remedy Partners and

University of Oklahoma Tulsa conduct numerous projects [10,11] that use PDI to process healthcare data.

- CloverDX (CloverETL) [12] provides also visual control and automation of data flows.

On the contrary, the advantage of source-coded ETL solutions is that they give developers the flexibility to handle new requirements and create unit tests. However, theyrequire a certain level of programming and maintaining skills in the development team and cause burdens on the overall project. Java and Python are common languages for ETL developers: Java is a general-purpose, object-oriented programming language. It isgenerally faster and more efficient, because it is a compiled language. Spring Boot is an open-source Java-based framework that supports the creation of stand-alone, production-grade applications easily. Conversely, Python is a dynamically typed and interpreted language. Python is considered to have less learning curve than Java

2.2. ETL Processes

We collect three ETL processes with different technical requirements:

- ETL Process 1: Mapping MEONA database export in CSV to FHIR
- ETL Process 2: Mapping ORBIS HL7 to Template Data Documents (TDDs)
- ETL Process 3: Constructing SOAP envelopes with encoded TDD to openEHR via IHE XDS.b

Source Systems: Our data originates from 1) MEONA and 2) ORBIS:

1) MEONA [15] is a clinical software, which supports the entire medication treatment process. MEONA offers: Safe medication prescription and administration, and efficient documentation of measures and reports at the push of a button.
2) ORBIS [15] is a solution for workflows in medicine, administration and management of healthcare facilities. UKSH currently uses a customized version of HL7 v.2, which is the most widely implemented healthcare standard for electronic data exchange in the clinical domain, to exchange data to and from ORBIS.

Target Systems: HAPI FHIR JPA server [16] is a FHIR server that uses relational databases and handles all storage and retrieval logic. Apart from HAPI FHIR, there are other different FHIR servers in the market: Blaze, Firely Server, IBM FHIR Server, etc. Additionally, Better Platform is a commercial Clinical Data Repository (CDR), which is designed to store and manage EHRs based on openEHR specifications.

In case of ETL Process 1, we aim to map information from the MEONA database as exported CSV files into FHIR R4 resources. As part of the MMI use case POLAR_MI, we detect medication problems and use medical data to optimize clinical care. The data are exported as CSV files. The result is loaded to a self-hosted HAPI FHIR server and later will be evaluated by analysis scripts from the POLAR_MI consortium. By ETL Process 2 (part of the HiGHmed use case Cardiology [2]), we attempt to detect decompensation in heart failure patients by mapping a mixture of standard HL7 messages (ADT andBAR) and customized ORU reports into Template Data Documents (TDDs) [11]. TheseTDDs can be consumed by the ETL process described in ETL Process 3. In ETL Process3, we construct SOAP envelopes with encoded TDDs to “IHE XDS.b Affinity Domain in conjunction of openEHR clinical data repository [13]”.

3. Implementation

ETL Process 1 is developed with Java and Spring Boot. Its ETL components are organized within a three-layered pattern [17]: the application layer, domain layer and data access layer. Each layer provides services to the next higher layer. The application layerexecutes the desired thread. In the domain layer, we define the source entities and the mapping processes with help of the HAPI FHIR library. The infrastructure layer is responsible for the data access and delivery of FHIR resources.

To fulfil the mapping of ETL Process 2, two ETL tools, Nifi and Talend, are connected as a "broker pattern [17]" using the streaming platform Apache Kafka. Nifi, due to its extendable customize feature, is in charge of receiving pipe-formatted HL7 messages, aggregating a unique identifier in the form of a Master Patient Identifier (MPI) [3] and converting the HL7 format to XML-based. Subsequently, Nifi pushes the messages to Kafka for distributing messages on different topics, whereas the Kafka topics are named after the HL7 message type. Finally, each Talend job reads from its corresponding topic, maps the HL7 data into openEHR templates and enriches it by querying and adding terminologies from a terminology server.

In ETL Process 3, we construct SOAP envelopes with Talend in a pipe-filter pattern [17]. We divide our process into multiple "filters" e.g. querying the XDS Value Sets from the terminology server and encoding the result TDDs from ETL Process 2. Talend's GUI offers a more visible way to map multiple XDS metadata and create "filters" here.

4. Lesson Learned

Three important components in a general ETL process in healthcare data integration are introduced: the ETL tools, the data origin and common healthcare data exchange formats. Besides giving this overview of the current healthcare data exchange area, we demonstrate three ETL processes, using different ETL tools to satisfy the data conversion requirements of our MeDIC into the resources in introduced formats.

Data from different sources and the target output formats strongly affect the decision of the correct ETL tools. Not all the requirements can be handled by a single ETL tool, every ETL tool has its advantages and limitations. For instance, ETL Process 1 utilizes the Java library *HAPI FHIR Structures FHIR R4* to create the expected output classes of FHIR resources. Similarly, ETL process 2 can be implemented in the same way using Java. Using Talend to map from HL7 to openEHR proved to be complex. In that context, we encountered during the development of ETL Process 2 the problem that the target XML templates can be too large for the Talend engine to handle.

Another problem we faced was the uneven skill disparity in the team. The challenge for our developers was the understanding of both fields, software development and healthcare data. For instance, some clinical experts are more familiar with standard terminologies like SNOMED CT and LOINC. On the other hand, technical experts focus more on infrastructure communication and software engineering. This problem can be also solved by choosing and combining the right tools. For the non-source code analysts and developers, the graphical components of Talend, like *tXMLMap*, provide visible and easy-to-understand pipelines. At the same time, Nifi is a good medium point, as it allows us to extend its functionalities by creating new source-code-based custom processors flexibly; while having a relatively simple graphical interface to manage the ETL pipelines. Hence, we use both Talend and Nifi together in ETL Process 2. We use

Talend to handle complex healthcare data transformations without source code and developadditional functionality with Nifi.

In addition to choosing the right ETL tools for developers, well-designed architecture patterns can also increase the level of abstraction and thus the team efficiency and workload management. ETL Process 3 shows the strengths of a pipe filter pattern by reusing "filters" to query "Value Set" from a terminology server and be able to process incoming TDD files from multiple processes.

In summary, the decision on the appropriate ETL tools and architecture patterns can save laborious work and time; however, it depends on the project purpose and skill of the development team.

5. Conclusion

Appropriate ETL tools and architecture patterns enable the development of healthcare data exchange efficiently, thereby advancing medical informatics research in general. Choosing the right tools and architecture pattern is critical, because once an architecture is in place, it is very hard (and expensive) to change. The shown implementations with tools and architecture patterns are adaptable for further ETL projects with a similar purpose. Nevertheless, further topics like data provenance, virtualization, unit tests and evaluation should also be in consideration for ETL processes.

6. Contribution of Authors

KC and SP developed the approach and drafted the manuscript.BS designed the high-level MeDIC architecture and gave methodological input. All authors read and approved the final manuscript.

7. Acknowledgement

This research was funded by the German Federal Ministry for Education and Research (BMBF, grant number 01ZZ1802T).

References

[1] Gehring S, Eulenfeld R. German Medical Informatics Initiative: Unlocking Data for Research and Health Care. Methods Inf Med. 2018 Jul;57(S 01):e46-e49. doi: 10.3414/ME18-13-0001. Epub 2018 Jul 17. PMID: 30016817; PMCID: PMC6178201.

[2] Haarbrandt B, Schreiweis B, Rey S, Sax U, Scheithauer S, Rienhoff O, Knaup-Gregori P, Bavendiek U, Dieterich C, Brors B, Kraus I, Thoms CM, Jäger D, Ellenrieder V, Bergh B, Yahyapour R, Eils R, Consortium H, Marschollek M. HiGHmed - An Open Platform Approach to Enhance Care and Research across Institutional Boundaries. Methods Inf Med. 2018 Jul;57(S 01):e66-e81. doi: 10.3414/ME18-02-0002. Epub 2018 Jul 17. PMID: 30016813; PMCID: PMC6193407.

[3] Mullin R. ETL Challenges within Healthcare Business Intelligence - Health IT. [Internet]. [cited 2022Aug.30]. https://www.healthitanswers.net/etl-challenges-within-healthcare-business-intelligence/.

[4] Singh A, Singh J. A comparative Review of Extraction, Transformation and Loading Tools. Database Systems Journal. 2018 Jun 23.

[5] Kherdekar VA, Metkewar PS. A technical comprehensive survey of ETL tools. International Journal of Applied Engineering Research. 2016;11(04):2557. doi: 10.37622/IJAER/11.4.2016.2557-2559
[6] Dhaouadi A, Bousselmi K, Monnet S, Gammoudi MM, Hammoudi S. A multi-layer modeling for the generation of new architectures for Big Data Warehousing. Advanced Information Networking and Applications. 2022;:204–18.
[7] Katragadda R and Sremath Tirumala S and Nandigam D. ETL tools for Data Warehousing: An empirical study of Open Source Talend Studio versus Microsoft SSIS; 2015 Jan 18.
[8] McPadden J, Durant TJ, Bunch DR, Coppi A, Price N, Rodgerson K, Torre CJ Jr, Byron W, Hsiao AL, Krumholz HM, Schulz WL. Health Care and Precision Medicine Research: Analysis of a Scalable Data Science Platform. J Med Internet Res. 2019 Apr 9;21(4):e13043. doi: 10.2196/13043. PMID: 30964441; PMCID: PMC6477571.
[9] Mai PL, Sand SR, Saha N, Oberti M, Dolafi T, DiGianni L, Root EJ, Kong X, Bremer RC, Santiago KM, Bojadzieva J, Barley D, Novokmet A, Ketchum KA, Nguyen N, Jacob S, Nichols KE, Kratz CP, Schiffman JD, Evans DG, Achatz MI, Strong LC, Garber JE, Ladwa SA, Malkin D, Weitzel JN. Li-Fraumeni Exploration Consortium Data Coordinating Center: Building an Interactive Web-Based Resource for Collaborative International Cancer Epidemiology Research for a Rare Condition. Cancer Epidemiol Biomarkers Prev. 2020 May;29(5):927-935. doi: 10.1158/1055-9965.EPI-19-1113. Epub 2020 Mar 10. PMID: 32156722; PMCID: PMC7196512.
[10] Miller R, Coyne E, Crowgey EL, Eckrich D, Myers JC, Villanueva R, Wadman J, Jacobs-Allen S, Gresh R, Volchenboum SL, Kolb EA. Implementation of a learning healthcare system for sickle cell disease. JAMIA Open. 2020 Oct 23;3(3):349-359. doi: 10.1093/jamiaopen/ooaa024. PMID: 33215070; PMCID: PMC7660956.
[11] Simplified Data Template (SDT). [cited 2022Aug30]. Available from: https://specifications.openehr.org/releases/ITS-REST/latest/simplified_data_template.html
[12] CloverDX. How cloverdx helps healthcare companies manage complex data [Internet]. CloverDX. [cited 2022Aug30]. Available from: https://www.cloverdx.com/blog/ cloverdx-healthcare-companies-manage-complex-data
[13] Wettstein R, Merzweiler A, Klass M, Heinze O. Using openEHR in XDS.b Environments - Opportunities and Challenges. Stud Health Technol Inform. 2020 Jun 26;272:300-303. doi: 10.3233/SHTI200554. PMID: 32604661.
[14] Cross-Enterprise Document Sharing (XDS.b): [Internet]. IHE ITI TF Vol1. [cited 2022Aug30]. Available from: https://profiles.ihe.net/ITI/TF/Volume1/ch-10.html
[15] Ebbing L, Kunze T, Scherließ R. Studien zur Arzneimitteltherapiesicherheit: Einführung einer Verordnungssoftware und Kompatibilitätsstudien von Insulin human mit anderen Parenteralia. Kiel: Universitätsbibliothek Kiel; 2018.
[16] HAPI FHIR - The Open Source FHIR API for Java. [Internet]. Smile CDR. [cited 2022Aug30]. Available from: https://hapifhir.io/
[17] Richards M. Software Architecture Patterns. California: O'Reilly Media, Inc.; 2015. ISBN 978-3938912073 (english)