

FML2Mirth: Generic Transformation of complex laboratory data based on FML mapping rules and automatically generated Mirth channels

Jesse Kruse, Joshua Wiedekopf, Ann-Kristin Kock-Schoppenhauer, Andrea Essenwanger, Josef Ingenerf, Hannes Ulrich

Submitted to: JMIR Medical Informatics
on: February 20, 2024

Disclaimer: © The authors. All rights reserved. This is a privileged document currently under peer-review/community review. Authors have provided JMIR Publications with an exclusive license to publish this preprint on its website for review purposes only. While the final peer-reviewed paper may be licensed under a CC BY license on publication, at this stage authors and publisher expressly prohibit redistribution of this draft paper other than for review purposes.

Table of Contents

Original Manuscript..... 5

Supplementary Files..... 17

 Figures 18

 Figure 1..... 19

 Figure 2..... 20

 Figure 3..... 21

 Multimedia Appendixes 22

 Multimedia Appendix 1..... 23

FML2Mirth: Generic Transformation of complex laboratory data based on FML mapping rules and automatically generated Mirth channels

Jesse Kruse¹ MSc; Joshua Wiedekopf^{2, 3} MSc; Ann-Kristin Kock-Schoppenhauer³ Dr rer hum biol; Andrea Essenwanger⁴ MSc; Josef Ingenerf^{2, 3} Dr rer nat; Hannes Ulrich⁵ Dr rer nat

¹LADR Laboratory Group Dr. Kramer & Colleagues Geesthacht DE

²Institute of Medical Informatics University of Luebeck Luebeck DE

³IT Center for Clinical Research University of Luebeck Luebeck DE

⁴mio42 LLC Berlin DE

⁵Institute for Medical Informatics and Statistics Kiel University and University Hospital Schleswig-Holstein Kiel DE

Corresponding Author:

Hannes Ulrich Dr rer nat

Institute for Medical Informatics and Statistics

Kiel University and University Hospital Schleswig-Holstein

Kaistraße 101

Kiel

DE

Abstract

Background: Reaching a significant amount of interoperability between proprietary healthcare systems is an ubiquitous task in medical informatics, where communication servers are traditionally used for referring and transforming data from source to target systems. The Mirth Connect Server, which is an open-source communication server, offers in addition to the exchange functionality, also functions for the simultaneous manipulation of data. In recent times, the standard Fast Healthcare Interoperability Resources (FHIR) is becoming more and more prevalent in the German healthcare system. This standard specifies its own standardized mechanisms for transforming data structures in the form of StructureMaps and the FHIR Mapping Language.

Objective: In this study, a generic approach will be developed, which allows to apply these formalized mapping rules defined by the FHIR Mapping Language in an exchangeable manner. A transformation engine is required to execute the mapping rules.

Methods: FHIR natively defines resources to support conversion of instance data, the FHIR resource StructureMap. This resource encodes all information required to transform data from a source system into a target system. In our approach, this information is defined in an implementation-independent manner using the FHIR Mapping Language. Once the mapping has been defined, executable Mirth channels are automatically generated from the resources containing the mapping in form of JavaScript. These channels can be deployed to the Mirth Connect Server.

Results: The resulting tool is called FML2Mirth, a Java-based transformer that derives Mirth channels from a given mapping based on the underlying StructureMaps that contain the detailed rules. The implementation of the translate functionality is guaranteed by the integration of a terminology server and to achieve conformity with existing profiles, validation via the FHIR validator is built in. The system is evaluated for its practical use by transforming LDTv.2 laboratory results into MIO lab reports according to NASHIP specifications and into the HL7 Europe Laboratory Report. It is shown that the system can generate complex structures, but LDTv.2 lacks some information to fully comply with the specification.

Conclusions: The tool for the auto-generation of Mirth channels was successfully presented. Initial tests have shown that it is feasible to use the complex structures of the mapping language in combination with a terminology server to transform instance data. Although the Mirth Server and FHIR are well established in the field of medical informatics, the combination offers space for more research, especially with regard to the FHIR Mapping Language. At the same time, it can be stated that the Mapping Language still has implementation short comings that can be compensated by Mirth Connect as a base technology. Clinical Trial: N/A

(JMIR Preprints 20/02/2024:57569)

DOI: <https://doi.org/10.2196/preprints.57569>

Preprint Settings

1) Would you like to publish your submitted manuscript as preprint?

✓ **Please make my preprint PDF available to anyone at any time (recommended).**

Please make my preprint PDF available only to logged-in users; I understand that my title and abstract will remain visible to all users.

Only make the preprint title and abstract visible.

No, I do not wish to publish my submitted manuscript as a preprint.

2) If accepted for publication in a JMIR journal, would you like the PDF to be visible to the public?

✓ **Yes, please make my accepted manuscript PDF available to anyone at any time (Recommended).**

Yes, but please make my accepted manuscript PDF available only to logged-in users; I understand that the title and abstract will remain visible to all users.

Yes, but only make the title and abstract visible (see Important note, above). I understand that if I later pay to participate in <http://www.jmir.org/>

Original Manuscript

Original Paper

Jesse Kruse^{1, 2}, Joshua Wiedekopf^{2,3}, Ann-Kristin Kock-Schoppenhauer³, Andrea Essenwanger⁴, Josef Ingenerf^{2,3}, Hannes Ulrich^{3,5}

1. LADR Laboratory Group Dr. Kramer & Colleagues, 21502 Geesthacht, German
2. Institute of Medical Informatics, University of Luebeck, 23562 Luebeck, Germany
3. IT Center for Clinical Research, Lübeck, University of Luebeck, 23562 Lübeck, Germany
4. mio42 LLC, 10587 Berlin, Germany
5. Institute for Medical Informatics and Statistics, Kiel University and University Hospital Schleswig-Holstein, Campus Kiel, Kiel and Lübeck, Schleswig-Holstein, Germany.

FML2Mirth: Generic Transformation of complex laboratory data based on FML mapping rules and automatically generated Mirth channels

Abstract

Background: Reaching meaningful interoperability between proprietary healthcare systems is a ubiquitous task in medical informatics, where communication servers are traditionally used for referring and transforming data from source to target systems. The Mirth Connect Server, an open-source communication server, offers in addition to the exchange functionality, functions for the simultaneous manipulation of data. The standard Fast Healthcare Interoperability Resources (FHIR) is recently becoming more and more prevalent in national healthcare systems. FHIR specifies its own standardized mechanisms for transforming data structures using StructureMaps and the FHIR Mapping Language (FML).

Objective: In this study, a generic approach is developed, which allows to apply declarative mapping rules defined by the FHIR Mapping Language in an exchangeable manner. A transformation engine is required to execute the mapping rules.

Methods: FHIR natively defines resources to support conversion of instance data, such as FHIR StructureMap. This resource encodes all information required to transform data from a source system into a target system. In our approach, this information is defined in an implementation-independent manner using the FHIR Mapping Language. Once the mapping has been defined, executable Mirth channels are automatically generated from the resources containing the mapping in form of JavaScript. These channels can then be deployed to the Mirth Connect Server.

Results: The resulting tool is called FML2Mirth, a Java-based transformer that derives Mirth channels from detailed declarative mapping rules based on the underlying StructureMaps. The implementation of the translate functionality is provided by the integration of a terminology server and to achieve conformity with existing profiles, validation via the FHIR validator is built in. The system is evaluated for its practical use by transforming LDTv.2 laboratory results into MIO lab reports according to NASHIP specifications and into the HL7 Europe Laboratory Report. It is shown that the system can generate complex structures, but LDTv.2 lacks some information to fully comply with the specification.

Conclusions: The tool for the auto-generation of Mirth channels was successfully presented. Our tests have shown the feasibility to use the complex structures of the mapping language in combination with a terminology server to transform instance data. Although the Mirth Server and FHIR are well established in the field of medical informatics, the combination offers space for more research, especially with regard to the FHIR Mapping Language. At the same time, it can be stated

that the Mapping Language still has implementation short comings that can be compensated by Mirth Connect as a base technology.

Trial Registration: N/A

Keywords: FHIR; StructureMaps; FHIR Mapping Language; Mirth; Laboratory Data



Introduction

The digital change is progressively transforming healthcare systems, and the rapid pace of digitalization is creating new clinical data sources that need to be integrated. This is of enormous importance for patient care, as data integration and fusion of all sources allow a comprehensive, holistic overview and ensure the best treatment possible. However, the lack of interoperability of healthcare systems is a significant and long-lasting problem [1]. Still, interoperability is required to ensure seamless and effortless access to essential healthcare information. Existing medical data is presented in a multitude of proprietary formats and different standards [2]. Therefore, it is inevitable to transform data into a harmonized structure to enable a collective use. Current (inter)national initiatives are integrating large volumes of data in clinical environments to enable their use [3–5]. The established data integration is close to the data origin which is highly desirable since a late mapping harbors disadvantages and issues can be addressed accordingly [6].

Aligning the multitude of standards and thus making data usable is a major task. The mapping from source to target data structures is mainly defined by data stewards and followed by a qualitative evaluation involving medical professionals. The involvement of these key professional groups is much easier if concise, declarative mapping rules are used instead of less comprehensible program instructions in various scripting languages. Basically, the creation of mapping rules is time-consuming and demands a significant number of resources. It occurs at every site involved, which commits a substantial additional number of stewards and medical professionals across the entire healthcare system [6,7]. Furthermore, sharing and reusing of declarative mapping rules in a standardized and adaptable manner is highly desirable and can release the needed resources and speed up the data enabling. For these reasons, the presented study aims to provide a software solution called FML2FHIR that is based on separating the declarative mapping rules as such from a transformation pipeline leveraging the functionalities of Mirth.

Methods

The transformation of healthcare-related information needs two major components to enable generalization: a highly adaptable exchange format and a modular transformation engine. Our approach is based on the HL7 standard Fast Healthcare Interoperability Resources (FHIR) and the Mirth communication server.

HL7 FHIR and the FHIR Mapping Language

HL7 FHIR is the emerging standard for healthcare-specific data exchange and finds broad adaption around the globe [8]. FHIR provides state-of-the-art technologies to modernize the current healthcare landscape using extensible resources as harmonized and semantically annotatable information units [9]. But FHIR also provides mechanisms to natively define transformations on its structures. The essential FHIR resource for automated conversion of instance data is the *StructureMap*, which defines a mapping from a source structure to a target structure [10] and provides all necessary information for an automatic transformation. To aid the definition of StructureMap resources, FHIR has specified a domain-specific, declarative language, the FHIR Mapping Language (FML) [11] which was introduced as part of FHIR Release 3. By specifying the mappings in an implementation-independent manner, the mapping between structures can be easily shared within the medical

informatics community.

One major benefit is the usage of generic data structures as input formats, since FML can define mappings on non-FHIR structures. Consequently, the FML can also be used to map older healthcare-related formats such as HL7 v.2 or LDT to FHIR.

The declarative syntax of the FML enables users to define mappings concisely. The mapping itself is structured in so called *groups* within the mapping, being formalized by mapping rules. These rules consist of two parts separated by an arrow. On the left side of the arrow is the source part. Here, fields of source structures can be accessed, and the values can be written in variables. On the right side, the fields of a target structure can be accessed and built-in transformation functions can be invoked (see Figure 1). Those provided functions cover a broad functionality, ranging from simple *copy* till a *translate* function to resolve given concept codes using an external FHIR *ConceptMap*.

```
group populateObservation(source test : LDT_Test, target observation : Observation) {
  // snip..

  test.LDT_8410 as s_code -> observation.code as code,
    code.coding = translate(s_code, "<URL>", "coding") "set Observation code";
  // Rule 1

  test.LDT_8420 as s_result -> observation.valueQuantity as quantity,
    quantity.value = s_result then {
  // Rule 2

  test.LDT_8421 as s_unit -> quantity.unit = translate(s_unit, "<URL>", "code") as t_unit,
    quantity.system = "http://unitsofmeasure.org",
    quantity.code = t_unit
    "set unit";
  } "set result";
  // Rule 3

  // snip..
}
```

Figure 1 The code illustrates an FML snippet showing how a conventional laboratory test in LDT format can be mapped to an FHIR observation resource. The mapping is structured in groups, containing the specific rules.

Mirth Connect Communication Server

Mirth Connect is an open-source communication server designed specifically for the healthcare sector. It features support for domain-specific formats, such as HL7v.2, as well as more general formats such as XML and JSON. Mirth is widely used as a communication server and is applied in various areas for data integration [12,13]. As a communication server, Mirth receives messages from one system and forwards them to downstream systems. Mirth introduces the concept of Channels to establish a connection from a source system to various target systems. Within the channels rule-based transformers can be defined to enable message manipulation. The received message can be automatically modified in terms of content or structure. The transformer rules are implemented as JavaScript code snippets which Mirth applies to the messages. The channels including its transformers are serialized as XML files and can easily be shared and redeployed at other sites.

Architecture for Generic Data Transformation

The envisioned two-folded process, shown in Figure 2, shall provide a light and adaptable data transformation: transformation design (1) and staging (2).

During the transformation design phase, the source and target systems are selected, and the mapping is declared in FML. The mapping needs to be created manually by a data-steward. This process can be supported by previously proposed algorithms and tools [14]. In addition, semantic concepts and codes can be mapped from source to target based on prepared mapping relationships provided by the

FHIR ConceptMap resource, if necessary. It is recommended that this process is done by a healthcare professional to ensure data validity and integrity.

In the staging phase, the Mirth channel, which was built based on the prior mapping efforts, is injected in the productive communication server at runtime. The mapping within the transformer uses the rules derived from FML for the format manipulation and dynamically translates the semantic codes using the ConceptMap.

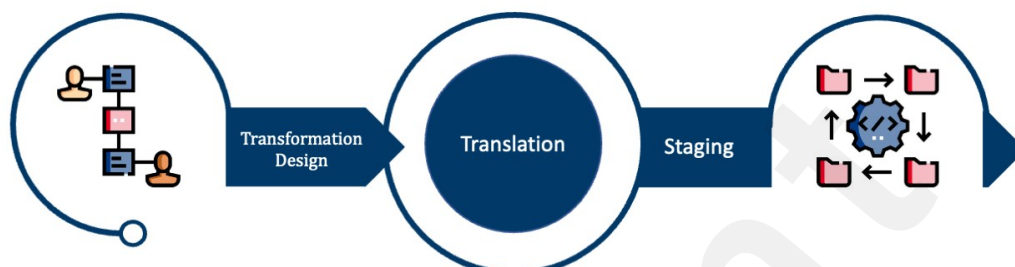


Figure 2 Conceptual Overview of the two-step process. In the first phase, the mapping is formalized as a FML script and corresponding semantic CodeSystems are aligned in a ConceptMap to each other. In the second phase, the priorly defined rules are used to deploy the mapping to Mirth. If semantic translation is needed the ConceptMaps can be queried during runtime.

Results

The proposed system implements the transition between the two process phases transformation design (1) and staging (2). This transition is achieved by the Java-implemented generator *FML2Mirth* [15], which derives Mirth channels from a given mapping. There are two kinds of files used in the process that need to be created upfront. For once, there are the ConceptMaps used by a terminology server to provide the translation service for concepts that need to be translated during the transformation of data in the production phase. These ConceptMaps are referenced by the *translate* operations in the FML to specify how a given concept in the source structure should be mapped to a concept used in the target structure.

In the second file, the FML script, the mapping between source and target structure is defined. This script is then used as an input for *FML2Mirth*. As a first step the FML-Script must be parsed, for which we use the official FHIR validator, since it provides functionality to generate a StructureMap from a given FML-Script. An additional advantage of this approach is, that by using a standard tool, it is guaranteed that the produced artifact is a standard-compliant StructureMap. Using these files as the input, *FML2Mirth* should also be compatible with every other StructureMap that adheres to the standard.

Besides the FML script, *FHIR2Mirth* needs the API address of the Mirth server to which the generated channel should be deployed, as well as the address of the terminology server that is queried for the translation service during a transformation in the production phase. The tool then parses the StructureMap and generates the JavaScript code needed for every transformation specified. Once the generation is finished, the resulting code is embedded within a Mirth channel definition and deployed on the specified server via the Mirth REST-API. The incoming messages are automatically transformed into the desired structure. On encounter of the FML *translate* function, Mirth queries the given terminology server to perform translation between the different CodeSystems of the source and the target structure based on the previously created ConceptMaps. In our setting shown in Figure 3, a HAPI FHIR JPA server instance is used as an external terminology server, but any other FHIR terminology server implementation could be used instead. The transformed message

is then forwarded to the defined target system, e.g., a FHIR repository.

The evaluation of the transformation within Mirth is implemented using the JUnit testing framework. While JUnit is most commonly used for unit testing, it can also be used for end-to-end integration testing. The JUnit runner loads LDT lab messages from files and sends them to the Mirth server via the REST API. The destination connector of the test channel is a JavaScript connector that returns the transformed message as a response to the requesting system. The JUnit runner then receives the transformed message and uses HAPI FHIR to parse the incoming bundles as FHIR resources. Thus, the incoming resources are automatically tested for their conformity to the specification and validated against the corresponding profiles using the HAPI instance validator.

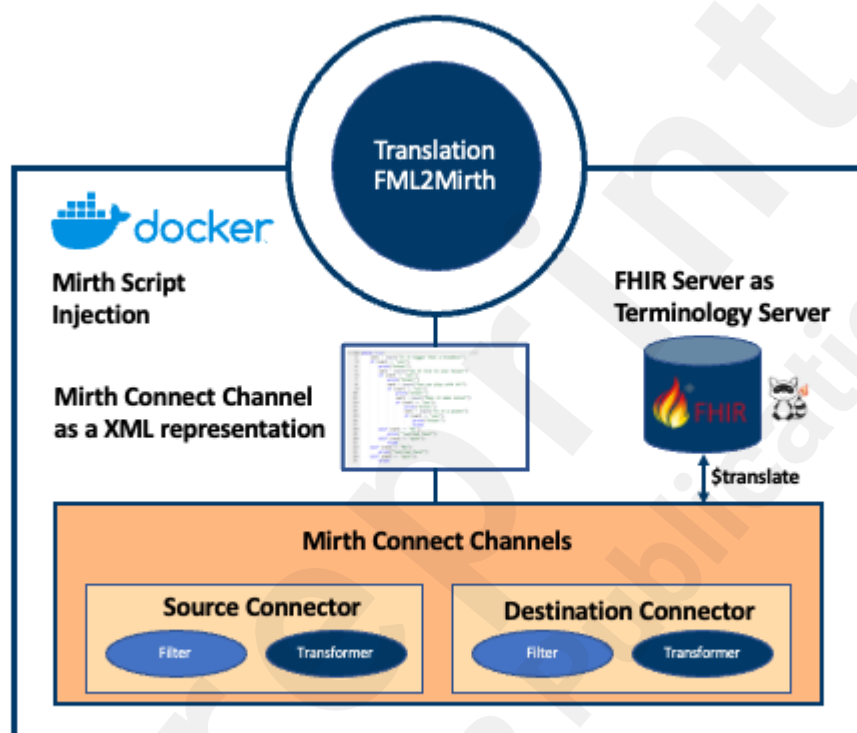


Figure 3 Conceptual overview of the proposed architecture for a generic data transformation process.

Evaluation

The assessment of our proposed transformation process is carried out in cooperation with the LADR laboratory network (LADR) [16], an association of 19 specialist laboratories throughout Germany serving more than 400 clinics. As an evaluation scenario, laboratory findings standardized in the German *Labordatenträger* (LDT) [17] format are transformed into FHIR resources. The LDT format is a perennial standard developed and published by the National Association of Statutory Health Insurance Physicians (NASHIP, in German: Kassenärztliche Bundesvereinigung, KBV) and is used daily for communicating up to one million messages in Germany. In our evaluation scenario, LADR provided 20 anonymized LDT laboratory messages representing complete blood counts in LDT version 2 according to the data format description in the FHIR version 5.12 for the evaluation. Two different FHIR Implementation Guides were used as target formats: the German Laboratory finding of the NASHIP and the HL7 Europe Laboratory Report [18].

The LDT format is mapped to both target formats using FML. Using the FML2Mirth generator, the FML scripts are translated into StructureMaps, and two separate channels are generated and

deployed on a Mirth instance. Each format has a distinct set of semantic codes, which are permissible values for certain elements (binding). The sending system in our evaluation scenario uses proprietary codes that must be mapped to the specified CodeSystem in the target profiles. A terminology server is used to provide a translate service based on ConceptMaps which is accessed by the Mirth channel via a REST API. The initial FML script for the transformation to the target format *MIO Laborbefund* with its nine profiles [19] of the NASHIP consists of over 400 lines of formalized mapping rules, resulting in StructureMaps with over 3500 lines and more than 2300 elements. The 20 transformed LDT laboratory findings resulted in an average of 62 unique FHIR resources each. Only one report failed the closing instance validation due to the missing first name within the corresponding patient resource.

The second Mirth channel transforming LDT to the eight profiles of the Europe Laboratory Report is based on the initial FML script. The same 20 LDT laboratory results sum up a total of over 694 FHIR resources, resulting in an average of 35 resources each. The subsequent validation process revealed for all test cases a structural issue that occurred due to a required extension of the DiagnosticReport profile. The extension should be a Composition reference to the report, but the extension itself is missing in the provided FHIR package by EU lab report, and the referral link is broken. Yet, a manual check and a comparison with the NASHIP MIO resources proved their validity.

Discussion

This study presented a concept and implementation of an adaptable transformation process for various data structures into a standardized format. Our approach is based on FHIR resources as a standardized data exchange format, FML as the declarative transformation rules and Mirth Connect as the transformation executor.

Principal Results

The approach focusses on separating the transformation rules from the actual transformation to extract the rules out of the process. Due to the detachment, the rules, which are created in a labor-intensive process, can be shared in a declarative and standardized manner. The use of FML for defining the mappings can cover technical operations like the network calls to the terminology server and are triggered by FML2Mirth dynamically during the JavaScript generation. In addition, the separation allows a degree of modularization and thus the reuse of the created rules. For example, it was possible to modify the mapping for the MIO reports to generate laboratory reports, that conform to the profiles of the EU laboratory report within a few hours. This emphasizes the flexibility of the proposed method. A significant advantage of this approach is that it allows mappings of concepts to be exchanged at runtime of the Mirth channel and developed and updated independently of the structural mapping. Compared to common ETL jobs, this approach needs much less manual intervention.

The mapping is defined using FML to render the StructureMap using the official FHIR Validator. The declarative mapping is enclosed in the generated StructureMap instances as a parse tree. Referring and combining various FML scripts is possible per the specification, but not implemented yet. Therefore, the mapping shall be edited incrementally during the transformation design in the FML scripts rather than in the StructureMaps directly. Nevertheless, the initial creation of the FML scripts is less time-consuming but still a labor-intensive task and currently there is a lack of suitable tool support. A FML script is always a one-way-mapping, and the reverse transformation is not implied automatically.

The functionality provided by Mirth highly constrains the design of our approach, as external dependencies should be kept to a minimum to ensure ease of use. With this in mind, the implemented

single command line tool in Java injects the transformations into a specified Mirth channel. In addition, only the Mirth server and a FHIR server, which provides terminology services, are required. At the same time, the system should be as flexible as possible and support arbitrary StructureMaps. To fulfill these requirements, it was necessary to work with the circumstances given by Mirth. Since Mirth itself works on XML objects, it makes sense to use this approach as well. So, the transformations defined in the StructureMap are translated into transformations on XML objects. Mirth itself first translates incoming data like HL7v2 to XML as a first processing step, so all formats supported by Mirth can automatically also be transformed using the generated transformer from FML2Mirth. If a format is not supported by Mirth, like LDT, simply another transformer step must be added, to transform the incoming data into an XML structure.

Limitations

During the evaluation, a problem was discovered with the FHIR implementation within the different FHIR servers. In particular, the issue concerned the FHIRPath implementation, which is used by the FHIR Validator to evaluate the validation rules. In our test setting, one rule is interpreted differently depending on the implementation used. On the tested laboratory reports, the .NET and the JavaScript implementations returned true, which is correct in our understanding. In contrast, both Java implementations tested (HAPI and IBM) returned false.

Alongside the complex structural mapping, semantic integrity must also be ensured. In our approach, the semantic mappings are created and validated by healthcare professionals and stored in a FHIR server as an external service – which is standard procedure. While existing terminology systems generally provide transition rules to newer versions, the maintenance of FHIR resources is not standardized and is an ongoing topic in the medical informatics research [20].

Furthermore, a mechanism is missing to process further StructureMaps, which are referenced from a given StructureMap. FML uses FHIRPath as an embedded language in several places. The associated specification is extensive and was implemented in this project only in parts.

Comparison with Prior Work

The use of formalized mapping rules in data integration is a well-studied topic, and Mirth as a transformation engine for clinical data integration is also a renowned tool. However, the combination of both topics with the goal of a generic and easily adaptive mapper is missing in the literature, to our knowledge. On the side of declarative rules, the work of Ong et al. [21] must be mentioned. The authors present an extensive study of a dynamic ETL approach that uses a custom mapping language to transform healthcare-related data into the OMOP common data model. The formalized rules are rich in details, but proprietary and rather database-oriented due to their use-case. An innovative feasibility study for FML has been presented by Dimitrov et al. [22]. The authors use FML for the transformation of the HL7 CDA based national Austrian electronic patient record. Using FML, a transformation from the CDA documents to the International Patient Summary (IPS) based on FHIR could be accomplished. The usage of Mirth as a transformation engine is an intuitive choice due to its functionality and product scope. So various studies are found dealing with the transformation from HL7 v2 message into further target formats, including structural reformation into JSON [23] or standardization into HL7 FHIR [24].

Our approach is based on a prior study that evaluated the transformation of StructureMaps into Mirth channels [25]. The results were promising, but it also emerged that the manual definition of StructureMaps became to be cumbersome and quickly very complex. The presented approach overcomes this shortage using FML and its intuitive nature.

Conclusions

We succeeded in implementing a functional system based on Mirth, which automatically generates transformations from StructureMaps and integrates them into a Mirth channel. The presented system supports large parts of the FML specification. First tests have shown that complex structures based on FML maps, generated in a channel in conjunction with a terminology server and ConceptMaps are feasible. If a new format is to be supported, it may only be necessary to insert another Mirth transformation step before the transformation from the StructureMap, to transform the input structure into XML. One advantage of the current solution is its flexibility and robustness in that only one resource is used for translation. The topic of FML is still an ongoing discussion and has not been extensively researched in the FHIR community so far. Accordingly, the tooling in this area is at this point, rather rudimentary.

Acknowledgements

This research was funded by German Federal Ministry of Education and Research, grant number 01ZZ2312A and 01ZZ2011.

Conflicts of Interest

N/A

Abbreviations

FHIR: Fast Healthcare Interoperability Resources

FML: FHIR Mapping Language

NASHIP: National Association of Statutory Health Insurance Physicians

IPS: International Patient Summaries

LDT: Labordatenträger

MIO: Medical Information Object (*Medizinisches Informationsobjekt*)

ETL: Extract, Transform, Load Process

OMOP: Observational Medical Outcomes Partnership

CDA: Clinical Document Architecture

References

1. Lehne M, Sass J, Essenwanger A, Schepers J, Thun S. Why digital medicine depends on interoperability. *Npj Digit Med* 2019 Aug 20;2(1):79. doi: 10.1038/s41746-019-0158-1
2. Ulrich H, Kock-Schoppenhauer A-K, Deppenwiese N, Gött R, Kern J, Lablans M, Majeed RW, Stöhr MR, Stausberg J, Varghese J, Dugas M, Ingnerf J. Understanding the Nature of Metadata: Systematic Review. *J Med Internet Res* 2022 Jan 11;24(1):e25440. PMID:35014967
3. Semler SC, Wissing F, Heyder R. German Medical Informatics Initiative. *Methods Inf Med*

2018 Jul;57(Suppl 1):e50–e56. PMID:30016818

4. Koch M, Richter J, Hauswaldt J, Krefting D. How to Make Outpatient Healthcare Data in Germany Available for Research in the Dynamic Course of Digital Transformation. *Stud Health Technol Inform* 2023 Sep 12;307:12–21. PMID:37697833
5. Auffray C, Balling R, Barroso I, Bencze L, Benson M, Bergeron J, Bernal-Delgado E, Blomberg N, Bock C, Conesa A, Del Signore S, Delogne C, Devilee P, Di Meglio A, Eijkemans M, Flicek P, Graf N, Grimm V, Guchelaar H-J, Guo Y-K, Gut IG, Hanbury A, Hanif S, Hilgers R-D, Honrado Á, Hose DR, Houwing-Duistermaat J, Hubbard T, Janacek SH, Karanikas H, Kievits T, Kohler M, Kremer A, Lanfear J, Lengauer T, Maes E, Meert T, Müller W, Nickel D, Oledzki P, Pedersen B, Petkovic M, Pliakos K, Rattray M, I Màs JR, Schneider R, Sengstag T, Serra-Picamal X, Spek W, Vaas LAI, van Batenburg O, Vandelaer M, Varnai P, Villoslada P, Vizcaíno JA, Wubbe JPM, Zanetti G. Making sense of big data in health research: Towards an EU action plan. *Genome Med* 2016 Jun 23;8(1):71. PMID:27338147
6. Kock-Schoppenhauer A-K, Schreiweis B, Ulrich H, Reimer N, Wiedekopf J, Kinast B, Busch H, Bergh B, Ingenerf J. Medical Data Engineering – Theory and Practice. In: Bellatreche L, Chernishev G, Corral A, Ouchani S, Vain J, editors. *Adv Model Data Eng Digit Era MEDI 2021 Commun Comput Inf Sci Vol 1481 Cham: Springer; 2021. p. 269–284. doi: 10.1007/978-3-030-87657-9_21*
7. Bönisch C, Kesztyüs D, Kesztyüs T. Harvesting metadata in clinical care: a crosswalk between FHIR, OMOP, CDISC and openEHR metadata. *Sci Data* 2022 Oct 28;9(1):659. PMID:36307424
8. Vorisek CN, Lehne M, Klopfenstein SAI, Mayer PJ, Bartschke A, Haese T, Thun S. Fast Healthcare Interoperability Resources (FHIR) for Interoperability in Health Research: Systematic Review. *JMIR Med Inform* 2022 Jul 19;10(7):e35724. PMID:35852842
9. Benson T, Grieve G, editors. *Principles of Health Interoperability: FHIR, HL7 and SNOMED CT. Fourth edition. Cham: Springer International Publishing; 2021. doi: 10.1007/978-3-030-56883-2_2 ISBN:978-3-030-56883-2*
10. HL7 International. *Reource StructureMap - FHIR v4.0.1. 2023. Available from: <http://build.fhir.org/structuremap.html> [accessed Aug 7, 2023]*
11. HL7 International. *FHIR Mapping Language - FHIR v4.0.0. 2023. Available from: <https://www.hl7.org/fhir/mapping-language.html> [accessed Oct 23, 2023]*
12. Camacho Rodriguez JC, Stäubert S, Löbe M. Automated Import of Clinical Data from HL7 Messages into OpenClinica and tranSMART Using Mirth Connect. *Stud Health Technol Inform* 2016;228:317–321. PMID:27577395
13. Lin J, Ranslam K, Shi F, Figurski M, Liu Z. Data Migration from Operating EMRs to OpenEMR with Mirth Connect. *Stud Health Technol Inform* 2019;257:288–292. PMID:30741211
14. Deppenwiese N, Duhm-Harbeck P, Ingenerf J, Ulrich H. MDRCupid: A Configurable Metadata Matching Toolbox. *Stud Health Technol Inform* 2019 Aug 21;264:88–92.

PMID:31437891

15. Kruse J. FML2Mirth. Zenodo; 2024. Available from: <https://doi.org/10.5281/zenodo.10678100> [accessed Feb 20, 2024]
16. LADR laboratory network. Available from: <https://www.ladr.de/> [accessed Feb 12, 2024]
17. Bundesvereinigung KK. LDT Labordatenträger Datensatzbeschreibung, LDT1001. 01 und Elektronisches Leistungsverzeichnis. Köln; 2004; Available from: https://update.kbv.de/ita-update/Labor/Labordatenkommunikation/EXT_ITA_VGEX_LDT%203_2_16_Gesamtdokument.pdf
18. HL7 FHIR. HL7 Europe Laboratory Report v0.1.0. Available from: <https://hl7.eu/fhir/laboratory/0.1.0-ballot/> [accessed Feb 12, 2024]
19. Laborbefund 1.0.0 - Laborbefund 1.0.0 - MIO. Available from: <https://mio.kbv.de/display/LAB1X0X0> [accessed Feb 20, 2024]
20. Wiedekopf J, Drenkhahn C, Rosenau L, Ulrich H, Kock-Schoppenhauer A-K, Ingenerf J. TerminoDiff - Detecting Semantic Differences in HL7 FHIR CodeSystems. Stud Health Technol Inform 2022 May 25;294:362–366. PMID:35612097
21. Ong TC, Kahn MG, Kwan BM, Yamashita T, Brandt E, Hosokawa P, Uhrich C, Schilling LM. Dynamic-ETL: a hybrid approach for health data extraction, transformation and loading. BMC Med Inform Decis Mak 2017 Dec;17(1):134. doi: 10.1186/s12911-017-0532-3
22. Dimitrov A, Duftschmid G. Generation of FHIR-Based International Patient Summaries from ELGA Data. Stud Health Technol Inform 2022 May 16;293:1–8. PMID:35592952
23. Cruz R, Guimarães T, Peixoto H, Santos MF. Architecture for Intensive Care Data Processing and Visualization in Real-time. Procedia Comput Sci 2021 Jan 1;184:923–928. doi: 10.1016/j.procs.2021.03.115
24. Alkarkoukly S, Kamal MdM, Beyan O. Breaking Barriers for Interoperability: A Reference Implementation of CSV-FHIR Transformation Using Open-Source Tools. In: Hägglund M, Blusi M, Bonacina S, Nilsson L, Cort Madsen I, Pelayo S, Moen A, Benis A, Lindsköld L, Gallos P, editors. Stud Health Technol Inform IOS Press; 2023. doi: 10.3233/SHTI230061
25. Vogl KM, Ulrich H, Ingenerf J. Generation of message transformers based on HL7 FHIR StructureMaps within the interface engine Mirth Connect. Infinite Science Publishing; 2020. p. 131–134.

Supplementary Files

Figures

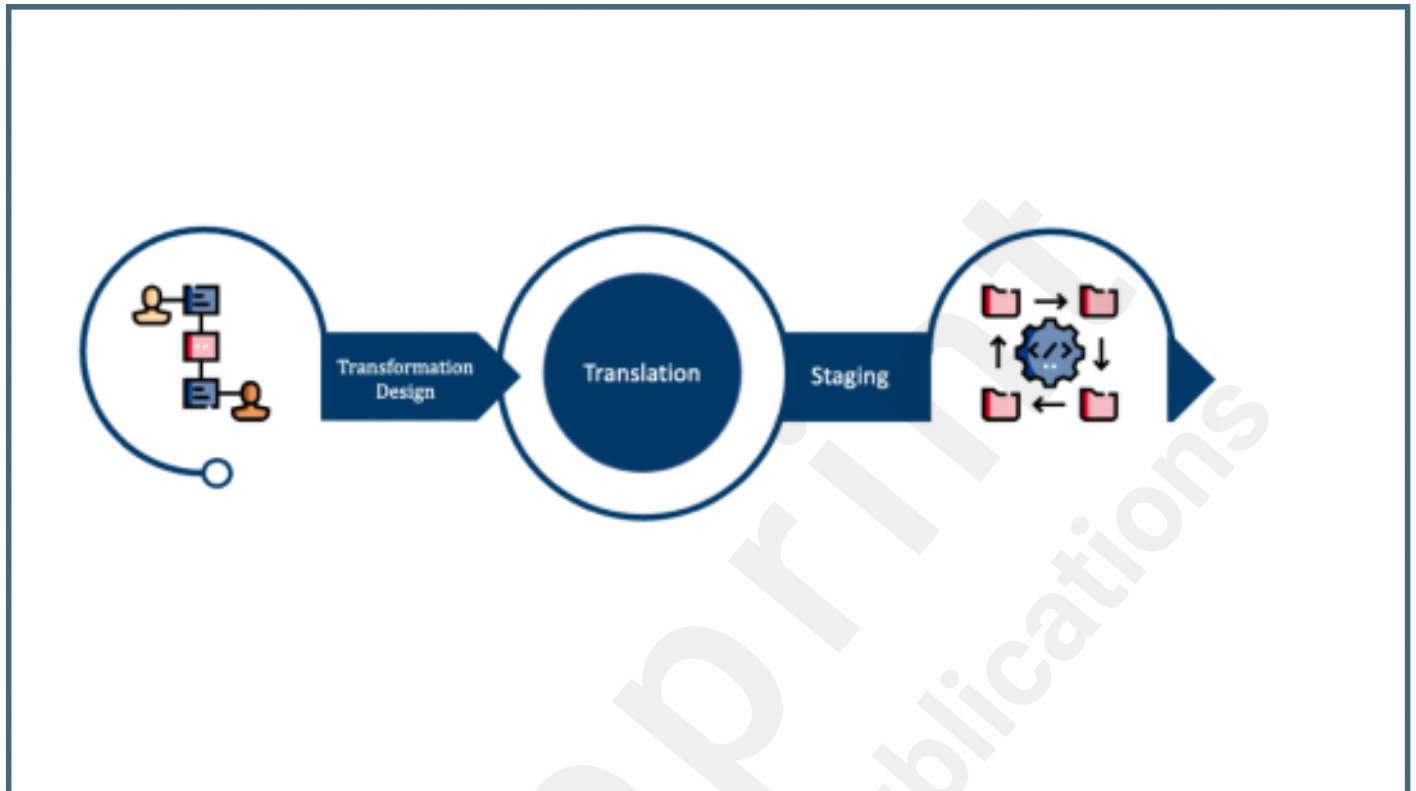
The code illustrates an FML snippet showing how a conventional laboratory test in LDT format can be mapped to an FHIR observation resource. The mapping is structured in groups, containing the specific rules.

```
group populateObservation(source test : LDT_Test, target observation : Observation) {
  // snip..

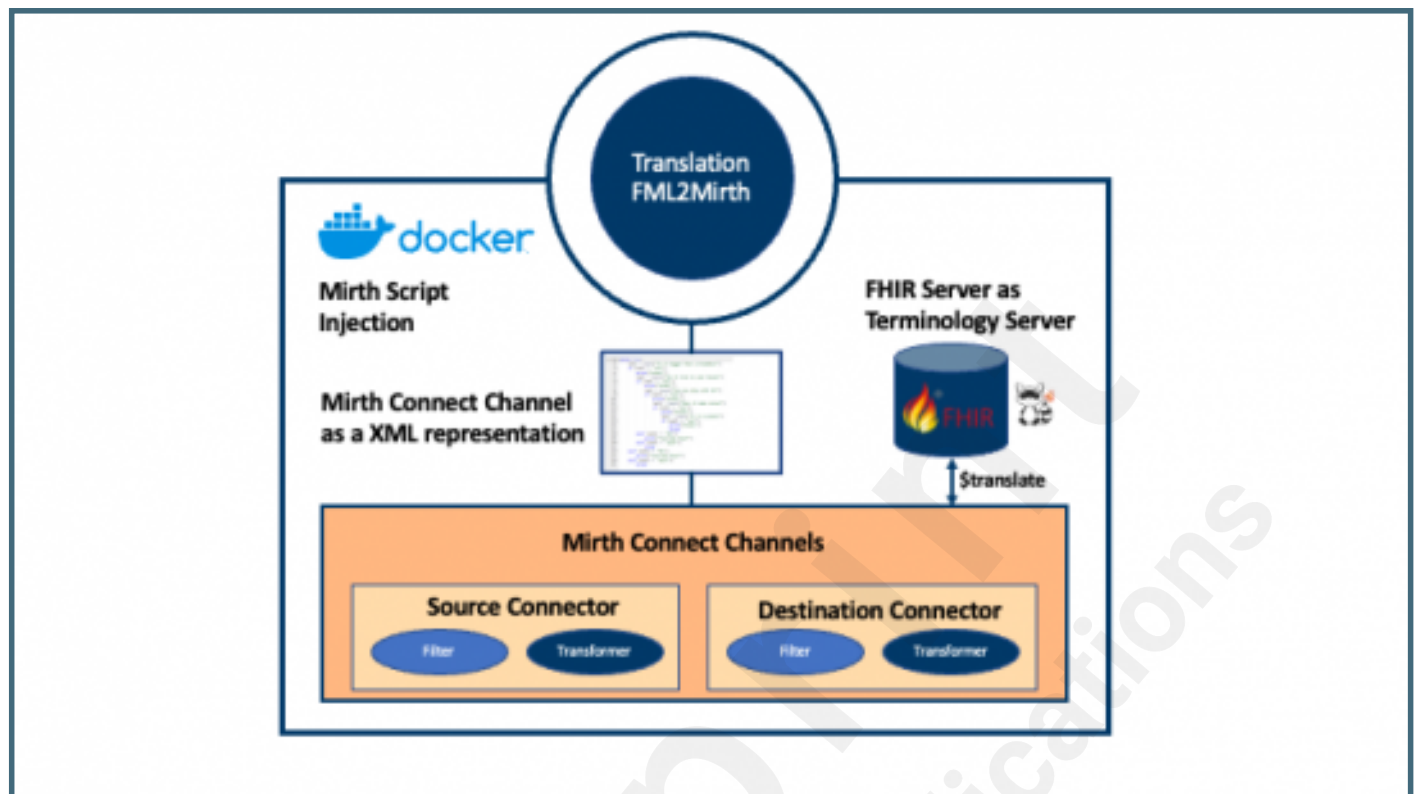
  test.LDT_8410 as s_code -> observation.code as code,
    code.coding = translate(s_code, "<URL>", "coding") "set Observation code";
  test.LDT_8420 as s_result -> observation.valueQuantity as quantity,
    quantity.value = s_result then {
  test.LDT_8421 as s_unit -> quantity.unit = translate(s_unit, "<URL>", "code") as t_unit,
    quantity.system = "http://unitsofmeasure.org",
    quantity.code = t_unit
    "set unit";
  } "set result";

  // snip..
}
```

Conceptual Overview of the two-step process. In the first phase, the mapping is formalized as a FML script and corresponding semantic CodeSystems are aligned in a ConceptMap to each other. In the second phase, the priorly defined rules are used to deploy the mapping to Mirth. If semantic translation is needed the ConceptMaps can be queried during runtime.



Conceptual overview of the proposed architecture for a generic data transformation process.



Multimedia Appendixes

This example shows the different transformation steps, including the FHIR StructureMap and JavaScript result.
URL: <http://asset.jmir.pub/assets/56496f4b01f2f5c24b0f0b667461909c.png>

