Challenges of Trustable AI and Added-Value on Health B. Séroussi et al. (Eds.) © 2022 European Federation for Medical Informatics (EFMI) and IOS Press. This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0). doi:10.3233/SHTI220475

TerminoDiff – Detecting Semantic Differences in HL7 FHIR CodeSystems

Joshua WIEDEKOPF^{a,b,1}, Cora DRENKHAHN^{a,b}, Lorenz ROSENAU^{a,b}, Hannes ULRICH^{a,b}, Ann-Kristin KOCK-SCHOPPENHAUER^b and Josef INGENERF^{a,b}

^aInstitute of Medical Informatics, University of Lübeck, Lübeck, Germany ^bIT Center for Clinical Research, University of Lübeck, Lübeck, Germany

Abstract. While HL7 FHIR and its terminology package have seen a rapid uptake by the research community, in no small part due to the wide availability of tooling and resources, there are some areas where tool availability is still lacking. In particular, the comparison of terminological resources, which supports the work of terminologists and implementers alike, has not yet been sufficiently addressed. Hence, we present TerminoDiff, an application to semantically compare FHIR R4 CodeSystem resources. Our tool considers differences across all levels required, i.e. metadata and concept differences, as well as differences in the edge graph, and surfaces them in a visually digestible fashion.

Keywords. Health Information Exchange; Terminology as Topic; Vocabulary; HL7 FHIR

1. Introduction

1.1. Background

The HL7 *Fast Healthcare Interoperability Resources* (FHIR) specification has seen an enthusiastic adoption by the Medical Informatics research community for the interoperable storage and exchange of medical information [1,2]. The enthusiastic uptake of this very recent specification is substantially due to the strong focus by the standards developers on the availability of tooling [1].

Parallel to the establishment of FHIR in healthcare IT, there is an increasing drive to incorporate standard terminology throughout the ecosystem, in order to maintain semantic interoperability of healthcare data across systems, institutional and even national boundaries. The terminological systems ICD-10, ICD-11, SNOMED CT and LOINC, among many others, play an important part in this development [1,3]. These terminological systems are maintained independently of the FHIR specification, but can be queried via the mechanism of FHIR terminology servers [3].

However, in real-world systems, standardized terminology has yet to fully supplant proprietary coding systems. Hence, the creation of interoperable terminology resources is an important part of ensuring healthcare interoperability on a whole; for which the

¹ Corresponding Author, Joshua Wiedekopf, Institute of Medical Informatics, University of Lübeck, Ratzeburger Allee 160, 23564 Lübeck, Germany; E-Mail: j.wiedekopf@uni-luebeck.de

HL7 FHIR Terminology Module provides a pathway that has seen adoption even outside of FHIR-compliant systems, like within the openEHR specification [4]. Additionally, to make use of the terminological knowledge contained in coding systems like ICD-10-GM, being the federally mandated classification for diagnoses within Germany, those resources need to be converted to FHIR directly [5].

There is tooling available both for the automatic and for the manual creation of terminology resources in FHIR [3,5]; however, little exists in the area of continuing maintenance of these resources. The development of these tools not only aids developers of terminologies, classifications and other artefacts, but also helps implementers of HL7 FHIR in migrating from one version to another.

While existing terminological systems generally provide some aids for transitioning to newer versions, such as history relationships in SNOMED CT or delta tables for ICD-10, maintenance of FHIR resources is not standardized, necessitating tooling support. Hence, we present a method and implementation for the computation and visualization of the differences between versions of the same HL7 FHIR CodeSystem.

1.2. Related Work

We have carried out a scoping literature review in order to get an overview of established algorithms in this field, searching the digital libraries *PubMed*, *Scopus* and *Springer Link* with the search string ("*fhir*" OR "ontology" OR "rdf" OR "terminology") AND ("diff" OR "change" OR "difference" OR "version").

In total, seven studies were considered applicable to the present work; more detail on this review is available online in our GitHub repository². Of these seven studies, none considered the maintenance of any kind of HL7 FHIR resources (including nonterminological resources like *Patient*); all studies were examining the problem from the view of formal ontologies and the semantic web.

The existing work in this closely related field [6,7] illustrates a clear need for the development of a difference computation. Additionally, the existing approaches generally do not provide a graphical (as in graph-based) view of the changes in connection between the versions, which we deem crucial for obtaining an understanding of relation between individual changes.

2. Methods

In HL7 FHIR, there are three kinds of terminological resources, namely *CodeSystem*, *ValueSet* and *ConceptMap*. We have focused on *CodeSystem* in this work, which is the resource type used for declaring the existence of a coding system with its associated metadata, and (generally) the concepts contained within that coding system with their associated relationships. There can be more than one kind of relationship, and polyhierarchical associations (where one concept may have more than one parent concept) are supported.

One example of a CodeSystem resource is one describing the currently 884 concepts of the OncoTree cancer classification, which are related in mono-hierarchical *parent*-relationships [8].

² https://itcr-uni-luebeck.github.io/TerminoDiff/SLR

Level	Aspect	Example	Resolution strategy
1	Metadata-level		Presentation as a table in the GUI
1.1	Simple differences	title, name, version	String comparisons
1.2	Differences within lists	identifier, language	(keyed) difference lists, e.g. by language.code
2	Concept-level		Presentation as a table in the GUI
2.1	Simple differences	display, definition	String comparisons
2.2	Differences within lists	property, designation	(keyed) difference lists, e.g. by <i>property.code</i>
2.3	Unilaterality of concepts	Deletions and additions of codes / concepts across versions	Surfacing in the table with dedicated filter and highlighting
3	Edge differences	Changes to properties linking concepts, i.e., <i>parent</i>	Creation and visualization of a difference graph

Table 1. Levels of differences we identified for FHIR R4 CodeSystem resources, and proposed resolution strategy

For obtaining a meaningful comparison of two CodeSystem resources, multiple levels must be considered, utilizing different strategies; for example, the metadata of the resource presents different challenges than the concept relationships. The levels of difference we identified for FHIR CodeSystem resources are provided in Table 1.

To illustrate the third level of differences, consider Figure 1, illustrating a difference graph for a fictitious pair of code graphs. While FHIR R4 allows at least three different approaches to specifying a *parent-child*-relationship (a property with code *parent*, one with code *child*, and the *concept* element within *concept*), we consider these to be semantically equivalent and reduce them to a canonical *parent* property. Our approach considers both the possibility of polyhierarchical relationships, where a concept may have more than one parent, as well as the possibility for multiple types of edges (such as *related-to* in this example).

3. Results

3.1. Implementation

The proposed software was implemented using the *Kotlin* programming language as a desktop application with a graphical user interface, utilizing the *Compose Desktop* toolkit. Our graph-based algorithms utilize the JGraphT library [9]

At the foundation of our implementation lie several components that build a difference model from the two provided CodeSystem resources. The user chooses one of these to be the *left* resource, the other the *right*, taking a view that is consistent with the side-by-side view commonly provided by generic comparison programs. Generally, the *right* resource should be chosen as a newer version of the *left* resource. For working with FHIR resources, we utilize the well-known *HAPI FHIR* library.

As the difference model must be constructed across several levels, as illustrated in Table 1, our engine also is split across multiple components. Metadata differences at Level 1 are shown to the user as a table; the order of items is chosen to be consistent to the FHIR specification of the CodeSystem resource.

Another component computes differences on the lists of concepts provided in each CodeSystem resource, surfacing values that are only referenced in one resource, as well as differences that occur in concepts referenced in each side.



Figure 1. Two fictitious code graphs representing two versions of a code system (left, right), and the difference graph (middle). In the difference graph, vertices and edges that only occur in the left code graph are shaded red dotted lines, those only in the right code graph are shaded green with solid edges. Dashed edges reduced to a *parent* relationship in this approach.

The third component renders a difference graph for the user. We compute a graph representing the additions and deletions of concepts and edges, illustrating the changes between the different versions. We do this in the same fashion as shown in Figure 1, allowing for a visual assessment of the changes within the CodeSystem hierarchy.

We make the source code of our program freely available via GitHub³ and Zenodo⁴ under the terms of the GPL 3.0 license.

3.2. Evaluation

The implementation was tested both with constructed examples, where the differences in metadata and/or concept relationships were well-known, as well as with real-world examples that were available in HL7 FHIR. For the latter, we utilized a number of examples, such as resources that have been specified in the context of the *GECCO* dataset by the German *Network University Medicine* [10], different versions of the *ICD-10-GM* classification, and the aforementioned *OncoTree* cancer classification [8].

Supporting our claim that our tool can improve the process of maintaining terminological artefacts, we were able to spot an omission in the OncoTree release notes (since corrected pursuant to our report), whereby a concept that has been introduced in version 2021-11-02 has not been referenced in the release notes.

As our software is still in active development, there has not yet been any formal evaluation of the tool with users not involved in software development; however, we are currently planning a survey among experts in terminology creation and maintenance that are active in the Medical Informatics Initiative in Germany and elsewhere.

4. Discussion

We believe that our implementation and framework can aid terminological authors in their day-to-day work of creation and maintenance, such that the tool can lead to greater acceptance and adoption of HL7 FHIR terminology and terminology servers at large

³ GitHub repository: https://github.com/itcr-uni-luebeck/TerminoDiff

⁴ Zenodo DOI: 10.5281/zenodo.5898267

scale. We believe this goal to be of supreme importance to the broader Medical Informatics research community to be able to ensure semantic interoperability across systems.

Furthermore, our approach could likely be amended to also consider other types of HL7 FHIR resources, such as the other two terminological resource types *ValueSet* and *ConceptMap*, but also other definitional artefacts, such as *StructureDefinition* (used for describing profiles of FHIR instance data), the maintenance of which we believe to benefit from additional tooling support as well.

5. Conclusion

The availability of user-friendly tooling is an important factor in the acceptance of standard in the industry. We have developed an application that tackles a challenge associated with developing new versions of terminological resources, as well as adopting these in applications. Doing so, we provide a tool that can help to increase the degree of semantic interoperability across the healthcare information landscape. In this fashion, we aid the transition from local in-house terminology to interoperable specifications, which is called for by the research community and political decision-makers alike.

Acknowledgements

This work was supported by the HiGHmed project within the Medical Informatics Initiative, funded by the Federal Ministry of Education and Research (grant number 01ZZ1802Z).

References

- [1] Benson T, Grieve G. Principles of Health Interoperability. Third edition. Springer International Publishing; 2016. https://doi.org/10.1007/978-3-319-30370-3.
- [2] Lehne M, Luijten S, Vom Felde Genannt Imbusch P, Thun S. The Use of FHIR in Digital Health A Review of the Scientific Literature. Stud Health Technol Inform 2019;267:52–8. https://doi.org/10.3233/SHTI190805.
- [3] Metke-Jimenez A, Steel J, Hansen D, Lawley M. Ontoserver: a syndicated terminology server. J Biomed Semant 2018;9:24. https://doi.org/10.1186/s13326-018-0191-z.
- [4] openEHR. Archetype Query Language (AQL) 2021. https://specifications.openehr.org/releases/QUERY/latest/AQL.html (accessed January 11, 2022).
- [5] Wiedekopf J, Drenkhahn C, Ulrich H, Kock-Schoppenhauer A-K, Ingenerf J. Providing ART-DECOR ValueSets via FHIR Terminology Servers - A Technical Report. Stud Health Technol Inform 2021;283:127–35. https://doi.org/10.3233/SHTI210550.
- [6] Hartung M, Groß A, Rahm E. COnto–Diff: generation of complex evolution mappings for life science ontologies. J Biomed Inform 2013;46:15–32. https://doi.org/10.1016/j.jbi.2012.04.009.
- [7] Klein M, Fensel D, Kiryakov A, Ognyanov D. Ontology Versioning and Change Detection on the Web. Knowl. Eng. Knowl. Manag. Ontol. Semantic Web, Springer Berlin Heidelberg; 2002, p. 197–212. https://doi.org/10.1007/3-540-45810-7_20.
- [8] Kundra R, Zhang H, Sheridan R, Sirintrapun SJ, Wang A, Ochoa A, et al. OncoTree: A Cancer Classification System for Precision Oncology. JCO Clin Cancer Inform 2021:221–30. https://doi.org/10.1200/CCI.20.00108.
- [9] Michail D, Kinable J, Naveh B, Sichi JV. JGraphT—A Java Library for Graph Data Structures and Algorithms. ACM Trans Math Softw 2020;46.
- [10] Sass J, Bartschke A, Lehne M, Essenwanger A, Rinaldi E, Rudolph S, et al. The German Corona Consensus Dataset (GECCO): a standardized dataset for COVID-19 research in university medicine and beyond. BMC Med Inform Decis Mak 2020;20:341. https://doi.org/10.1186/s12911-020-01374-w.