dHealth 2022
G. Schreier et al. (Eds.)

doi:10.3233/SHTI220351

TerminoloGit – An Open Source Terminology Server for Large Scale eHealth Environments

Gabriel KLEINOSCHEG[a,1], Nikola TANJGA[a], Nina SVEC[a], Stephan RAINER-SABLATNIG[a], and Stefan SABUTSCH[a]
[a] *ELGA GmbH, Vienna, Austria*

Abstract. Semantic interoperability is the centerpiece of successful communication between information systems within the health care system. As such, a terminology server provides information systems with the concepts needed for coding information accordingly. As the current Austrian terminology server does not meet all requirements anymore, a new terminology server is necessary. Consequently, after an unsuccessful call for bids, the Semantic Competence Center (SCC) of the ELGA GmbH set up a technologically more advanced solution for a new terminology server. The result is a Git based architecture in combination with the HL7® FHIR® IG publisher complemented by a FHIR® server and some in-house developments. The Git repository is hosted on GitLab.com, hence, allowing the use of GitLab's CI/CD functionalities for publishing the contents. Since the 1 January 2022 the new terminology server "TerminoloGit" is operational and provides 171 code lists and 141 value sets. Using open-source components only and making all parts of the system publicly available results in a cost-effective solution which may also be improved by the open-source community.

Keywords. Health Information Interoperability, Terminology, Clinical Coding, Health Level Seven, HL7, HL7 FHIR

1. Introduction

In order to achieve semantic interoperability throughout the healthcare environment, it is necessary that the information shared across different information systems is described using a common set of concepts [1]. These concepts are usually collectively provided in the form of code lists and value sets by a terminology server.

For the Austrian health care system such a terminology server has been active since 2013 [2]. This terminology server provides all basic functionalities which are required for the management and maintenance of terminologies such as administration of different versions of terminologies, providing various download formats (ClaML v2, extended SVS, csv), or SOAP web services. However, due to distributed responsibilities it would be cumbersome to add new functionalities such as support of other standards like the HL7® FHIR® standard or the connection to a FHIR® server. Furthermore, certain operations like upload and download of terminologies regularly resulted in timeouts especially when dealing with large terminologies.

[1] Corresponding Author: Gabriel Kleinoscheg, MSc, ELGA GmbH, Vienna, Austria, E-Mail: gabriel.kleinoscheg@elga.gv.at

As a result, the Semantic Competence Center (SCC) of ELGA GmbH initiated a European-wide call for bids regarding a new, state-of-the-art terminology server which would be used Austrian-wide. The main requirements among others for the new terminology server were stability, scalability, ease of use, and the support of current standards (e.g. HL7® FHIR®). Unfortunately, none of the bids were within the predefined budget which is why alternative technical solutions were sought.

Eventually, the SCC decided to opt for a Git [3] based solution in combination with the HL7 FHIR IG publisher [4], a self-hosted FHIR server [5], and some tools which have been developed in-house. The reasons for the decision were threefold. First, by using well-established and open-source software the operating costs of the new terminology server would diminish. Second, Git and FHIR represent the state-of-the-art in their respective realms which are software development and eHealth standards, respectively, both of which come with a very active community. Finally, the new terminology server ("TerminoloGit") should not only make the terminologies publicly available, but also the underlying implementation by making all developments open-source themselves.

The aim of this paper is to evaluate if the requirements for a new terminology server could be met with the architecture presented here.

2. Architecture

The architecture of TerminoloGit mainly comprises of the open-source components Git, GitLab, the HL7 FHIR IG publisher, and the FHIR server. They are complemented by the self-developed components Markup Language Converter for Clinical Terms (MaLaC-CT) and Implementation Guide Versioning (IGVer) (see Figure 1).

Git as distributed version control system allows easy management of different terminology versions by using branches, tags, etc. In order to make TerminoloGit publicly available and to leverage CI/CD functionalities, the project is hosted on GitLab.com. GitLab.com also offers the advantage that the system could easily be exported to a self-hosted GitLab CE or EE environment, if the requirement arises.

For the purpose of providing users with a browsable terminology server the IG publisher transforms all content into static HTML pages which are then published via GitLab Pages.

Additionally, all terminologies will be provided as FHIR CodeSystem and ValueSet resources on a self-hosted FHIR server ("FHIR-tx-at"). Based on functional and performance analysis the IBM® FHIR® Server [5] has been chosen.

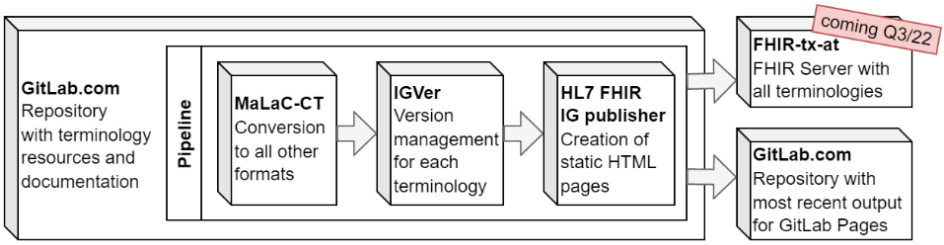

Figure 1. This figure illustrates the main components of the new terminology server "TerminoloGit" as well as the CI/CD pipeline.

2.1. MaLaC-CT

With the Markup Language Converter for Clinical Terms (MaLaC-CT) [6] the SCC developed a tool that makes it possible to provide users with several different file formats which are more or less commonly used to store and transmit terminologies. MaLaC-CT itself is an application written in Python which currently allows conversion of code lists and value sets between the formats specified in Table 1.

The HL7® FHIR® (XML and JSON) representation of code lists and value sets conform to the format standardized by HL7 (see [7] and [8], respectively). As such, these formats can be used to upload the terminologies onto a FHIR server.

FHIR Shorthand (FSH) on the other hand "is a domain-specific language for defining FHIR artifacts involved in the creation of FHIR Implementation Guides (IG)" [9]. By the time writing this paper version 1.0.0 was approved as a Standard for Trial Use (STU 1) and version 2.0.0 was available as "Mixed Normative-Trial Use" [10]. For processing FSH MaLaC-CT makes use of SUSHI which "converts FSH language into FHIR artifacts" [11].

The Classification Markup Language (ClaML) in version 2 [12] was already supported by the old terminology server, especially for code lists. For reasons of backwards compatibility version 2 is also supported by MaLaC-CT and subsequently by TerminoloGit. In addition, MaLaC-CT also implements the latest version of ClaML, version 3 [13].

The proprietary CSV format is a FHIR compatible CSV, which is primarily used to create or maintain a terminology e.g. in Excel.

In the old terminology server IHE's Sharing Value Sets (SVS) [14] format has been extended with some additional attributes for concepts by the ELGA GmbH. This format is primarily used for value sets.

Eventually, the outdated CSV conforms with the CSV format provided by the old terminology server.

Note, that the SVS and the outdated CSV can technically not contain all the information that is available in the other formats. These two formats are only available for legacy purposes.

Table 1. This table depicts all formats for code lists as well as value sets supported by MaLaC-CT.

Format	Version	File Type	File Extension
FHIR	4	XML	.4.fhir.xml
FHIR	4	JSON	.4.fhir.json
FHIR Shorthand	1	FSH	.1.fsh
FHIR Shorthand	2	FSH	.2.fsh
Classification Markup Language (ClaML)	2	XML	.2.claml.xml
Classification Markup Language (ClaML)	3	XML	.3.claml.xml
Proprietary CSV	1	CSV	1.propcsv.csv
Sharing Value Sets extended by ELGA	1	XML	.1.svsextelga.xml
Outdated CSV	1	CSV	.1.outdatedcsv.csv

MaLaC-CT automatically detects the provided file format based on the file's extension and subsequently creates all other file formats.

Apart from the conversion between the different formats MaLaC-CT also offers the possibility to upload a terminology to a specified FHIR server once the conversion has been completed.

2.2. IGVer

The Implementation Guide Versioning (IGVer) automatically detects which terminologies have been changed since the last successful pipeline run and preserves the old version so, it is accessible to users.

For this reason, the file names of all HTML pages related to an old terminology version will be prepended with the date when the old version has been committed to the repository.

Furthermore, IGVer prepares the download page of the respective terminology and (re-)sets Git tags named after the terminology for users to be easily informed when a terminology has changed.

2.3. TerminoloGit Template

Based on the IG template for Austria's Implementation Guides [15], a custom template for TerminoloGit has been developed [16]. Specifically, the presentation of code lists and value sets has been changed.

Instead of separate tabs for each file format (i.e. XML, JSON, etc.) the user will be provided with "Download" tab where all formats can be downloaded. With the intention of saving storage space in GitLab Pages, the source files of each format are not directly included into the IG publisher's output but the download links point directly to the repository hosted on GitLab.com (see Figure 2).

In addition, the tab "Previous Versions" enables the user to access all previous versions of the respective terminology and to compare the old version with the current one leveraging the diff-tool provided by W3C [17] (see Figure 3).

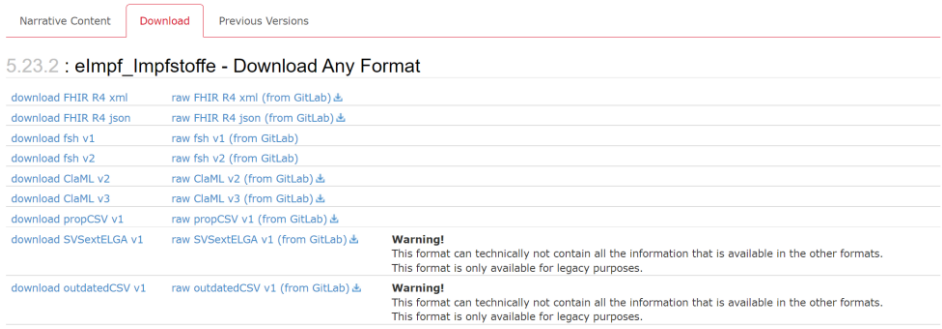

Figure 2. Presentation of a terminology in TerminoloGit. The tab "Download" contains links to all file formats.

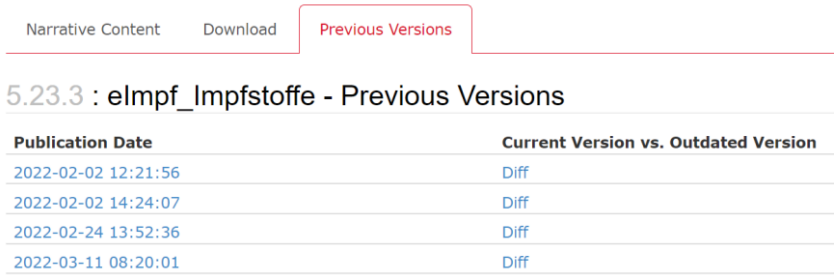

Figure 3. Presentation of a terminology in TerminoloGit. The tab "Previous Versions" allows access to old versions of the respective terminology.

3. Publication Process

The publication process for new or existing terminologies has been completely set up on GitLab Pipelines (see Figure 1). The pipeline will be invoked by a user who performs a Git commit, i.e., creating a new or updating an existing terminology.

Subsequently, the terminologies that have been added or changed since the last successful pipeline run will be passed on to MaLaC-CT to create or update all implemented file formats. In doing so, MaLaC-CT uploads the terminology to the FHIR® server as well.

As a next step, IGVer preserves the old version of changed terminologies, prepares the download pages and resets the Git tags. Once these steps are finished, the most recent version of the FHIR IG publisher is being downloaded and started, hence, creating static HTML pages for TerminoloGit. These are then pushed to a second GitLab repository where they are published as GitLab Pages.

4. Preliminary Results

As of 1 January 2022, TerminoloGit has gone public (see [18] and [19]). By then, 171 code lists and 141 value sets are hosted on TerminoloGit.

Currently, the FHIR server has not yet been made publicly available because it is not fully operational yet. For Q3/2022 it is planned to have the FHIR server stabilized and hardened in terms of availability and security, thus making it possible to access terminologies using FHIR's API.

Maintenance regarding terminologies will happen on both servers as long as the old terminology server is operational. Thus, software developers will have time to adapt their products to the new terminology server and to provide feedback regarding TerminoloGit probably resulting in improving the whole system.

5. Discussion

A terminology server for large scale eHealth environments mainly based on open-source components has been established. To meet the requirement of providing several file formats for terminologies and all old versions of terminologies some in-house development (MaLaC-CT and IGVer, respectively) was necessary. It has to be noted, that MaLaC-CT does not yet support proprietary delivery formats of terminologies (e.g. LOINC). As a result, these might still be missing on TerminoloGit.

Nevertheless, within less than a year a system has been set up which supports not only the publication and maintenance of terminologies in different file formats, but which has also proven to be adaptable (e.g. support of new file formats) with little effort and scalable by design (the system might easily be hosted on Docker containers). Furthermore, by using open-source and self-developed components only operational costs amount to a fraction of what the received bids would have proposed.

Maintenance of terminologies in TerminoloGit turns out to be far easier than on the old terminology server as the maintainer does not depend on a web interface to update terminologies any more but may updated terminologies in a local Git repository pushing the changes to the server as soon as all tasks have been accomplished. Furthermore, it was possible to design a transparent and comprehensible publication process, e.g.,

TerminoloGit allows users to compare old terminology versions with the current one. Moreover, by using Git it is easy to switch to an old version of the project in case of a mistake.

In order to make the knowledge gained by setting up TerminoloGit available for the broader public especially the eHealth community, all parts of the project have been made open-source. In this respect, active participation in further developments and improving the system are highly appreciated.

Please note that – as some parts are still under development – the processes described in this paper are subject to change.

Disclaimer: HL7®, and FHIR® are the registered trademarks of Health Level Seven International and their use of these trademarks does not constitute an endorsement by HL7.

References

[1] Rector A. L., Solomon W. D., Nowlan W. A., Rush T. W., Zanstra P. E., & Claassen W. M. (1994). A Terminology Server for Medical Language and Medical. *Methods of information in medicine, 34*(1-2), 147-57.
[2] *Terminologie-Browser.* (n.d.). Öffentliches Gesundheitsportal Österreichs. Retrieved 25 January 2022, from https://termpub.gesundheit.gv.at/TermBrowser/gui/main/main.zul
[3] *Git.* (n.d.). Git. Retrieved 25 January 2022, from https://git-scm.com/
[4] *GitHub - HL7/fhir-ig-publisher: Source code for the IG publisher*. (n.d.). GitHub. Retrieved 25 January 2022, from https://github.com/HL7/fhir-ig-publisher
[5] *IBM® FHIR® Server*. (n.d.). GitHub Pages. Retrieved 25 January 2022, from https://ibm.github.io/FHIR/
[6] *ELGA GmbH / MaLaC-CT*. (n.d.). GitLab. Retrieved 25 January 2022, from https://gitlab.com/elga-gmbh/malac-ct
[7] *Xml - FHIR 4.0.1*. (2019, November 1). Health Level Seven International. Retrieved 25 January 2022, from http://hl7.org/fhir/xml.html
[8] *Json - FHIR 4.0.1.* (2019, November 1). Health Level Seven International. Retrieved 25 January 2022, from http://hl7.org/fhir/json.html
[9] *HL7.FHIR.UV.SHORTHAND\FHIR Shorthand - FHIR v4.0.1. 1.0.0 - STU1* (2020, December 12). Health Level Seven International. Retrieved 25 January 2022, from http://hl7.org/fhir/uv/shorthand/
[10] *HL7.FHIR.UV.SHORTHAND\FHIR Shorthand - FHIR v4.0.1. 2.0.0 - Mixed Normative-Trial Use.* (2021, November 23). Health Level Seven International. Retrieved 25 January 2022, from http://build.fhir.org/ig/HL7/fhir-shorthand/
[11] *Documentation*. (n.d.). FSH School. Retrieved 25 January 2022, from https://fshschool.org/docs/
[12] *ISO 13120:2013*. (n.d.). ISO. Retrieved 25 January 2022, from https://www.iso.org/standard/52952.html
[13] *ISO Standards Maintenance Portal*. (n.d.). ISO. Retrieved 25 January 2022, from https://standards.iso.org/iso/13120/ed-2/en/
[14] *IHE ITI TF Vol1*. (n.d.). IHE. Retrieved 25 January 2022, from https://profiles.ihe.net/ITI/TF/Volume1/ch-21.html#21
[15] *GitHub - HL7Austria/at-fhir-ig-template: The base tempate for HL7 Austria Implementation Guides*. (n.d.). GitHub. Retrieved 25 January 2022, from https://github.com/HL7Austria/at-fhir-ig-template
[16] *ELGA GmbH / TerminoloGit Template*. (n.d.). GitLab. Retrieved 25 January 2022, from https://gitlab.com/elga-gmbh/terminologit-template
[17] *HTML Diff service.* (n.d.). W3C. Retrieved March 23, 2022, from https://services.w3.org/htmldiff
[18] *AUSTRIAN-TERMINOLOGY-SERVER\Home - FHIR v4.0.1. TerminoloGit* (2022, January 1). ELGA GmbH. Retrieved 25 January 2022, from https://termgit.elga.gv.at/index.html
[19] *ELGA GmbH / TerminoloGit Dev*. (n.d.). GitLab. Retrieved 25 January 2022, from https://gitlab.com/elga-gmbh/termgit