Governança de um repositório local baseado em openEHR em conformidade com openEHR Clinical Knowledge Manager / Governance of openEHR based local repository compliant with openEHR Clinical Knowledge Manager / Gobernanza de un repositorio local basado en openEHR en conformidad con openEHR Clinical Knowledge Manager

Vanessa Pereira, BSc [1,2,3]; Samuel Frade, MSc [2,3,4]; Leandro Gomes, MSc [3]; Tina Vavpotič, MSc [3]; Ricardo Cruz-Correia, PhD [1,2]

[1] Faculdade de Medicina da Universidade do Porto, Porto, Portugal

[2] CINTESIS – Centro de Investigação em Tecnologias e Serviços de Saúde, Porto, Portugal

[3] Marand d.o.o, Ljubljana, Slovenia

[4] BinaryScope Solutions, Funchal, Portugal

Correspondence to Vanessa Pereira, email: up201608624@med.up.pt

Abstract (EN)

Objectives: The aim of the study is to implement a suitable way of having the different *openEHR* local repositories updated and compliant with the *openEHR CKM* (Clinical Knowledge Manager).

Methods: A comparison was made between archetypes stored in a local repository of an ongoing project and the current version of the same archetypes in the *openEHR CKM*. A script for comparison of both repositories based on *Angular 2/Typescript* was made using *REST API* calls from *ADL Designer* and *GitHub API*.

Results: Creation of documentation about how to deal with the versioning of archetypes on new and ongoing projects, together with a script that runs on local repositories connected to *ADLDesigner* to search new archetype versions from *CKM*.

Conclusion: Contribution with an added value to the governance of *openEHR* resources by keeping them updated in a way that can facilitate and save time for the local repository management team, when searching new versions of *openEHR CKM*.

Keywords: (MeSH) Electronic Health Records, Medical Informatics, Health Information Exchange, (Non MeSH) OpenEHR, Version Control.

Resumo (PT)

Objetivos: O objetivo do estudo foi implementar uma forma adequada de ter os diferentes repositórios locais baseados em *openEHR* atualizados e em conformidade com o *openEHR CKM*.

Métodos: Foi feita uma comparação entre os arquétipos armazenados num repositório local de um projeto em progresso e a versão atualizada dos mesmos arquétipos no *openEHR CKM*. Um *script* de comparação de ambos os repositórios baseado em *Angular 2 / Typescript* foi criado, usando chamadas via *REST API* do *ADLDesigner e* do *GitHub*.

Resultados: Criação de documentação sobre como lidar com o controlo de versão de arquétipos em projetos novos ou em progresso, juntamente com um *script* que pode ser executado em repositórios internos conectados ao *ADL Designer* para pesquisar novas versões de arquétipos do *openEHR CKM*.

Conclusão: Contribuição para a governança de recursos do *openEHR,* mantendo-os atualizados de uma maneira que possa facilitar e economizar tempo para a equipa do repositório local quando pesquisar novas versões do *openEHR CKM*.

Palavras-chave: (MeSH) Registros Electrónicos de Saúde, Informática Médica, Troca de Informações em Saúde, (Non MeSH), OpenEHR, Controle de Versão.

1. INTRODUCTION

In a digital era, healthcare information had to adapt itself to the current technologies. A lot of new applications were developed and have improved the way physicians interact with electronic health records, but the underlying information model of how the patient data should be recorded and stored was left to be implemented exclusively by developers. To address this issue, some standards emerged over the years, and from those, the most promising is *openEHR* [1]. The international *OpenEHR* community has its own open governance library of archetypes and templates called Clinical Knowledge Manager (*CKM*) [2]. However, there is an emerging issue. Each company or institution working with *openEHR* archetypes and templates, cannot use it directly for each individual project since implementation specifications requires, very often, a fine-grained level of detail, very localized, which is not part of the internationally agreed upon common concepts. Also, system implementers need to have a working version of all artefacts used, so they need to have their own local repository with a file system or a control version system (*CVS*) to manage them. These local repositories are usually created at the start of a project by downloading copies of the available archetypes at a certain point in time and usually these are not kept updated. Over the years, the main *CKM* (also known as *International openEHR CKM*) had new updated versions for the different archetypes, having a very well-structured management of the archetype's lifecycles, with the new version being approved after a community consensus [3]. This implies that knowledge in the *openEHR CKM* stays always updated, but that does not necessarily happen on the local repositories.

Until now, there was no clear definition on how to solve this issue - how to manage a local repository, keep it connected to the CKM and maintain the local repository updated and compliant with it. Some companies have developed applications and modelling tools to work with the archetypes, like the *Archetype Editor* from Ocean Informatics and lately a *web-based* application, the *ADL Designer* [4] from Marand, which can make the web-connection with the different repositories based on *openEHR* artifacts and parse them.

OpenEHR

The *openEHR* is a free open-source standard which provides specifications of how to store, share and retrieve health data with the main idea of separating this data from applications implementation logic as an agnostic approach [5]. The *motto* of this standard is: "The transformation of health data from physical format to electronic format, thus ensuring universal interoperability between all forms of electronic data" [6]. In this case, the patient is the focus of the electronic health record (EHR). Some of the benefits of using this platform are the possibility of working the medical information at the same semantic level, strengthening the interoperability and making possible to use analytic functions as research querying and decision support. It is also possible to build an independent data repository for this EHR that does not need to know what kind of information will get in the beginning, because it can be modeled for what the healthcare institution requires as the needs appear. *OpenEHR* is divided in two-level modeling [7] that constitute the base information structure of an EHR: the first level, is composed by the Reference Model (RM) that gives the software specification and in the second level, the

Archetype Model (AM), where the archetypes and templates are presented. To understand what an archetype is, it is necessary to consider the basics of ontology, which are defined as “a set of concepts and categories in a subject area or domain that shows their properties and the relations between them” (Oxford dictionary, 2018). Thus, an archetype is the key feature of the separation between information models and domain models, that defines how to capture clinical data [8]. An archetype is composed by a set of data elements and other data types with formal definition of knowledge domain for a healthcare information system and can be shared and reusable in different situations, so it is not necessary to create new archetypes for different templates. These resources have a *Lego™* model approach, giving an elementary concept base to create structures - the templates.

OpenEHR CKM

Created in April 2009, a product from Ocean Health System (ex-Ocean Informatics) and under the management of the *openEHR* community, it has become the main web tool that makes the management of clinical models’ resources [9,10,11]. The first introduction of *openEHR CKM* had as main objective the creation of an archetype library, development of review processes with achieved content consensus, publication and governance of the artefacts. Since then, the possibility of adding terminology and other terminology specific subsets (e.g. *SNOMED CT, LOINC, ICD*) has been included. It offers a free registration for individuals from all around the world, focused on giving added value to the repository on a voluntary basis [12]. All non-technical healthcare area professions are also encouraged to contribute, it is not a requirement to be a physician to redound. It is possible to purpose new artifacts (archetypes and templates), suggest corrections and participate in discussions, translate archetypes to other languages, watch and adopt archetypes. All the changes to an archetype are subjected to a consensual decision from all reviewers before being published. The *openEHR CKM* has a model governance system that supports all the life cycle of archetypes, templates and terminology. Inasmuch as there are different requirements between different countries or even projects, other instances of CKM for national registries have been made, such as *Norwegian Nasjonal IKT CKM*, *Australian Digital Health Agency CKM*, *NHS England CKM*, *Shared UK CKM (Apperta Foundation, Scottish Government)* and the *Slovenian MoH CKM* [13].

2. METHODS

The aim of the study is to find and implement a suitable way of having the different *openEHR* local repositories updated and compliant with the international *openEHR CKM*. In order to achieve it, the following studies were made:

1. Learn how the *openEHR CKM* works, history, features, functionalities and how the verification of new versions of artifacts is made;
2. Study state of art methodologies used for managing software and document lifecycles;
3. Review of the *openEHR* specifications for Reference Model (RM) and Archetype Model (AM), related with archetype identification and versioning;
4. Creation of documentation about how to deal with archetype versioning on a local repository and a *script* to compare the artifacts content in both repositories and verify new updates for each archetype from *openEHR CKM*;
5. Development and implementation of the proposed methodology and verification of results.

The junction of all the information gathered from these studies allowed to create a methodology and script with rules for versioning comparison. Initially a comparison was made, between archetypes stored in a local repository of an ongoing project and the current version of the same archetypes in the international *openEHR CKM,* and afterwards between both versioning parameters. The provided local repository with artifacts was hosted at a *GitLab service*. It is a private version control repository that needs authentication to access the information within. The *REST API* provided by this service was in version v4. The comparison repository, *openEHR CKM,* is publicly accessible and is hosted on *GitHub*. A *REST API* is also provided by this service and authentication is not necessary to access to repositories content, because it is public. This repository is a mirror of all the content available at *http://www.openehr.org/ckm/*. To test the *REST API* calls in both repositories Postman was used, which is a tool for prototyping and testing *HTTP Calls*, with a lot of other testing features and friendly graphical user interface (*GUI*). To get archetypes from repositories and expose them in a create and edit environment, it was used *ADLDesigner* and some REST API calls available from its features. The *script* will be executed from within this web tool. For the *script* development it was used the *Visual Studio Editor* from *Microsoft* and the code for the script was written in *Angular 2/Typescript*, a variant from *JavaScript*.

Identifying parameters of new versions of archetypes

OpenEHR has dedicated a section that specifies the versioning of archetypes and templates, called the Archetype Object Model (AOM)[14], a package inside of the Archetype Model (AM) that contains many classes like *ARCHETYPE, ARCHETYPE_HRID* and *AUTHORED_ARCHETYPE* with version parameters [15,16]. An archetype has two ways of being identified, by human readable identification (*HRID*) or in a machine-readable identification. In the *HRID* it is possible to divide the archetype name in two ways, in *semantic_id* with the versioning extension of an under-development archetype and the *physical_id*:

- *Semantic_id*, composed by:

(*namespace::rm_publisher-model_name-rm_class.concept_id.v.major_version.minor_version.patch_version-version_status.build_count)*:

  - **org.openehr::openEHR-EHR-OBSERVATION.blood_pressure.v1.0.5-alpha**
  - **org.openehr::openEHR-EHR-OBSERVATION.blood_pressure.v1.0.5-rc.17**

- *Physical_id* , composed by:

(*namespace::rm_publisher-model_name-rm_class.concept_id.v.major_version.minor_version.patch_version*):

  - **org.openehr::openEHR-EHR-OBSERVATION.blood_pressure.v1.0.5**

In the case of machine readable way, these parameters are present on “*uid”*, “*MD5-CAM”*, “*build_uid”* and “*revision”*, which can be seen in the next excerpt from an Archetype Definition Language (ADL) file:

```
archetype(adl_version=1.4;uid=b1506a87-9bf2-4978-9eed-6ceecb0c2be9)
openEHR-EHR-OBSERVATION.blood_pressure.v1
lifecycle_state = <"published">
other_details = <
["MD5-CAM-1.0.1"] = <"7341F7E8A07ACE883A5F541BA79F2B95">
["build_uid"] = <"68040aab-98da-4b3c-a93c-c91df19b05f8">
["revision"] = <"1.1.1">
>
```

These parameters are under the "*other_details*" section and each one has a different content:

- “***uid***” is the unique identification number for each archetype. For example, the "blood pressure" archetype will always have the UID “b1506a87-9bf2-4978-9eed-6ceecb0c2be9”, even if the version of the archetype was changed. The format is made by an universal unique identifier (UUID), an identifier that is unique across both space and time.
- “***archetype_id***”: which contains the information about the archetype name including the major version, for example "blood pressure" is defined as “openEHR-EHR-OBSERVATION.blood_pressure.v1”, where v1 is the Major Version of this archetype.
- “***lifecycle_state***” which gives the stage of different life cycles associated to the archetype development (e.g. “published”).“
- “***MD5-CAM-1.0.1***” is a hash that is calculated by the actual values inside of the archetype, which means that every new version with changes will have a different hash code.
- “***build_uid***” - Every time the archetype gets a new change or version and is uploaded or committed to some repository, it will get a new "build_id", which is unique for every build. The format is also made by an UUID.
- “***revision***”: (e.g. “1.1.1”), based on the semantic versioning from *semVer* [17].

When an archetype is modified and gets a new version, the parameters that identify the versioning - *“revision”, “build_id”* and *“MD5-CAM”* - are also changed. These will be the parameters used for version comparison in the archetypes of both repositories.

3. RESULTS AND DISCUSSION

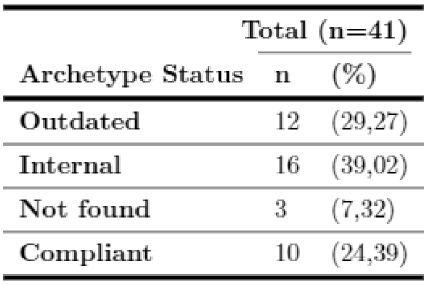
Analysis of archetype content from local repository and comparison with openEHR CKM repository

Before starting developing the script, an analysis of the provided repository was made. This analysis consisted on a direct comparison between the major version (V.x), versioning parameters and the content of archetypes inside of this local repository in *GitLab,* with the archetypes from the international *openEHR CKM* hosted on *GitHub*. The local repository had 41 archetypes. The primary search was made by archetype name *(archetype_id)* on both repositories. For each one of these archetypes was given a status to define the conformity with *openEHR CKM*:

- “**Outdated**": in case of versioning parameters (*Major Version, revision, archetype UID, MD5-CAM*) and content inside of the archetype being different in both repositories;
- "**Internal**", which is the case of creation of new archetypes that needs to meet certain clinical needs and are not present on the international openEHR CKM or when they are specialized to meet some local or national requirement;
- "**Not found**", if the archetype in the local repository was totally changed by another one on the international *openEHR CKM*, but with similar content, or even if this archetype does not exist anymore on the international openEHR CKM - case of archetype specializations that lost connection with the parent archetype;
- "**Compliant**", when the archetype has the same versioning parameters (*Major Version, revision, archetype UID, MD5-CAM*) and content are similar in both repositories, which means that has the latest version from the international CKM;

It was expected to have some outdated archetypes during the analysis, since it is a normal behaviour in the developing information systems - with time, changes are made, features added and *bugs* can be found. In the table 1 is possible to see the results of this comparison.

Table 1 - Summary of archetype comparison analysis from the provided CKM repository with *GitHub* mirror of *openEHR international CKM* (June 2018) (n=41)



	Total (n=41)	
Archetype Status	n	(%)
Outdated	12	(29,27)
Internal	16	(39,02)
Not found	3	(7,32)
Compliant	10	(24,39)

Although it is a repository with a high percentage of internal archetypes due to local or national requirements, which are not present in the international CKM, a lot of archetypes are outdated too. Some of them were downloaded to the local repository in November 2016 and since that date, there was no verification of new versions of archetypes from the international CKM. This can result in errors that can still exist in the local repository, that most probably were fixed on the international CKM, or even additional content that would enrich the local archetypes. Also, having a repository with these characteristics is not the aim of *openEHR* ideals, where less than ¼ of the artifacts are compliant with the international CKM. To manually verify same archetypes on both repositories, side by side and check all the content and versioning parameters took around nine hours to complete.

Current outcomes from openEHR artifacts usage

One of the aims of openEHR was to provide public archetypes, after being agreed by the international online community, in a single international repository (*openEHR CKM*) that could be used in several EHR systems around the globe. It is also possible to add regional or local archetypes that would still be compatible with the previous ones inserted in the international CKM (using CKM incubator feature) and making the information shareable with everybody. However, the current use of archetypes are not working in this way [18]. Each project that makes use of archetypes have their own repository. Some of them are using the international CKM and being compliant with it, others are not. Other projects have initially downloaded the archetypes from international *CKM*, but made a few changes to comply to specification requirements and then saved on local repositories like *GitHub, GitLab, SubVersion* or *Preforce*. This results in a jumble of archetypes variations from the international base of these archetypes, which was not the aim from *openEHR*.

Script definition

On the current settings it was defined that the typical user of the script to update the archetypes would be the person responsible for taking care of the governance of the local openEHR artifacts repository on in their respective project, role usually performed by a clinical modeler, health informatician or even a programmer (when access to the first two are not available). It starts with one of these users being the "CKM repository owner", opening and logging into in the "*ADL Designer*" web application, choosing the desired repository and entering in the artifacts page from that repository. A button "Check archetypes updates from OpenEHR CKM" on the top right of "ADL Designer" should be clickable. When the button is clicked, a new tab will be opened, and a script will run to check if there are new updates for the list of archetypes of that local repository, by fetching the list from GitLab REST API, and comparing them with the same archetypes in the GitHub repository of openEHR international CKM, using also the REST API service.

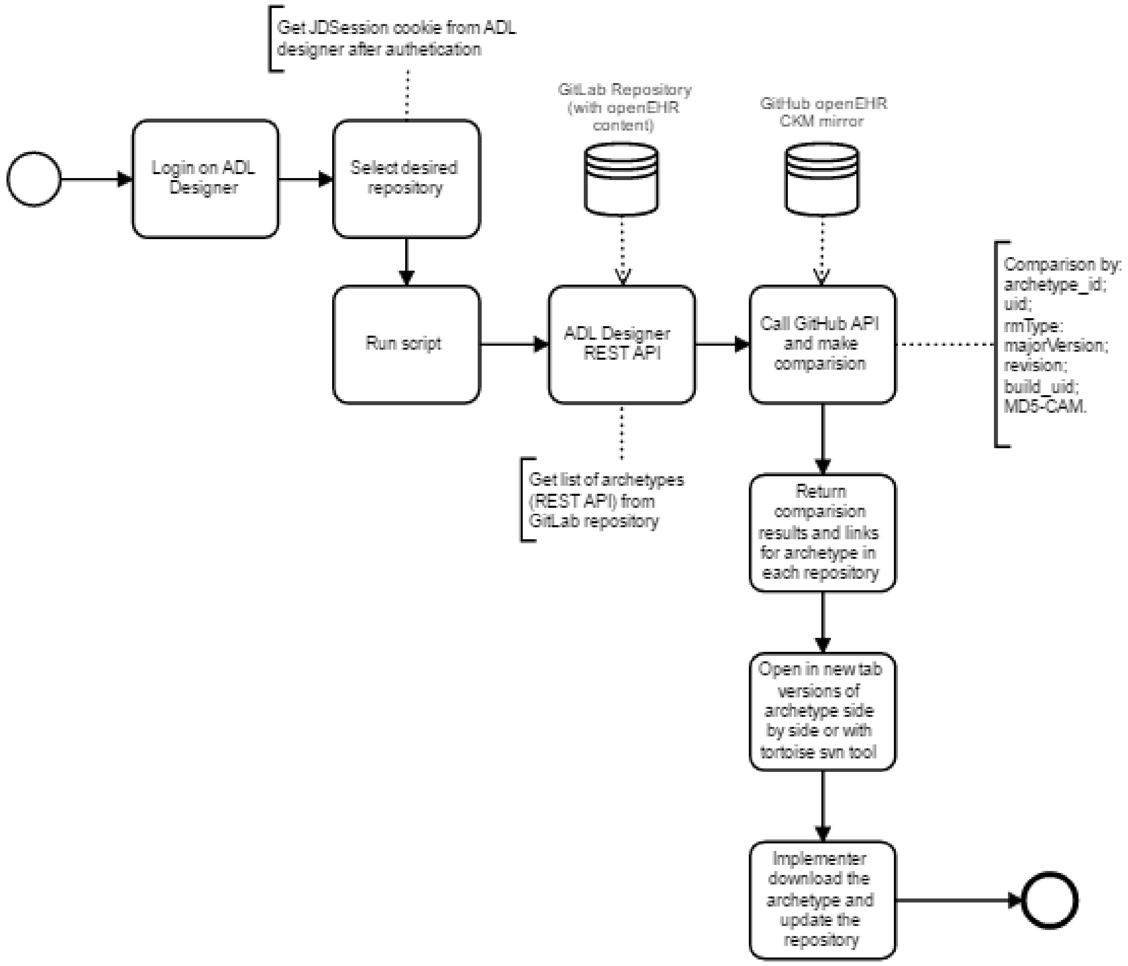
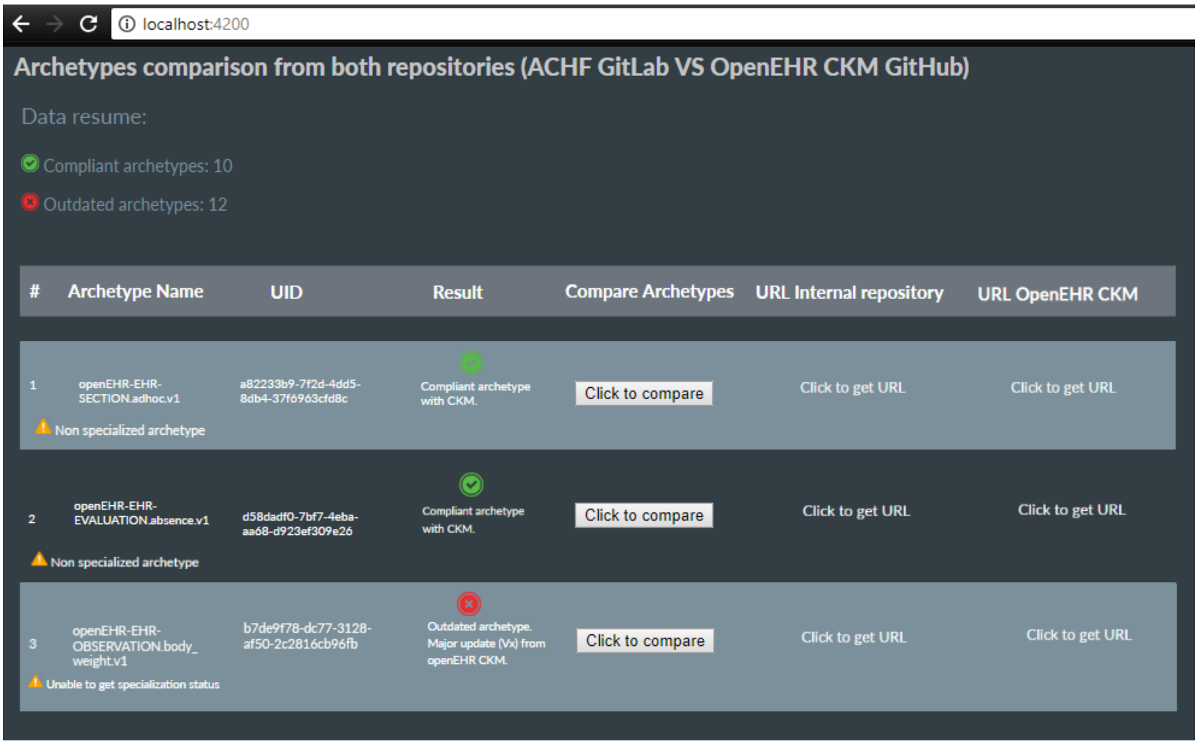

Get JDSession cookie from ADL designer after authetication
GitLab Repository (with openEHR content)
GitHub openEHR CKM mirror
Login on ADL Designer
Select desired repository
Run script
ADL Designer REST API
Call GitHub API and make comparision
Comparison by: archetype_id; uid; rmType: majorVersion; revision; build_uid; MD5-CAM.
Get list of archetypes (REST API) from GitLab repository
Return comparision results and links for archetype in each repository
Open in new tab versions of archetype side by side or with tortoise svn tool
Implementer download the archetype and update the repository

Figure 1 - Script process for archetype comparison

If there are updates, these results will appear in the web page (figure 2) and let the user know which archetypes have new versions. Clicking on the button “compare” in one of the archetypes, will open a new window in the browser and the user can compare both archetypes side by side, with highlighted lines in the changed content (figure 3), which allows the user to check what were the main changes in the new version quicker.

Then, if the user wants to download this new version of the archetype, a button to download it will be available under the comparison page.



localhost:4200

Archetypes comparison from both repositories (ACHF GitLab VS OpenEHR CKM GitHub)

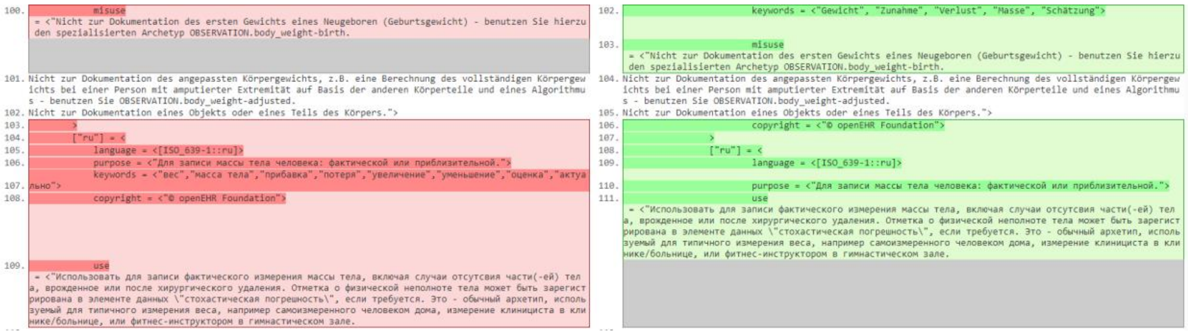
Data resume:

✓ Compliant archetypes: 10

✗ Outdated archetypes: 12

#	Archetype Name	UID	Result	Compare Archetypes	URL Internal repository	URL OpenEHR CKM
1	openEHR-EHR-SECTION.adhoc.v1 ⚠ Non specialized archetype	a82233b9-7f2d-4dd5-8db4-37f6963cfd8c	✓ Compliant archetype with CKM.	Click to compare	Click to get URL	Click to get URL
2	openEHR-EHR-EVALUATION.absence.v1 ⚠ Non specialized archetype	d58dadf0-7bf7-4eba-aa68-d923ef309e26	✓ Compliant archetype with CKM.	Click to compare	Click to get URL	Click to get URL
3	openEHR-EHR-OBSERVATION.body_weight.v1 ⚠ Unable to get specialization status	b7de9f78-dc77-3128-af50-2c2816cb96fb	✗ Outdated archetype. Major update (Vx) from openEHR CKM.	Click to compare	Click to get URL	Click to get URL

Figure 2 - Running script result showed on web page.



```
			misuse
 = <"Nicht zur Dokumentation des ersten Gewichts eines Neugeboren (Geburtsgewicht) - benutzen Sie hierzu den spezialisierten Archetyp OBSERVATION.body_weight-birth.
Nicht zur Dokumentation des angepassten Körpergewichts, z.B. eine Berechnung des vollständigen Körpergewichts bei einer Person mit amputierter Extremität auf Basis der anderen Körperteile und eines Algorithmus - benutzen Sie OBSERVATION.body_weight-adjusted.
Nicht zur Dokumentation eines Objekts oder eines Teils des Körpers.">
		>
		["ru"] = <
			language = <[ISO_639-1::ru]>
			purpose = <"Для записи массы тела человека: фактической или приблизительной.">
			keywords = <"вес","масса тела","прибавка","потеря","увеличение","уменьшение","оценка","актуально">
			copyright = <"© openEHR Foundation">
			use
 = <"Использовать для записи фактического измерения массы тела, включая случаи отсутсвия части(-ей) тела, врожденное или после хирургического удаления. Отметка о физической неполноте тела может быть зарегистрирована в элементе данных \"стохастическая погрешность\", если требуется. Это - обычный архетип, используемый для типичного измерения веса, например самоизмеренного человеком дома, измерение клинициста в клинике/больнице, или фитнес-инструктором в гимнастическом зале.
```

```
			keywords = <"Gewicht", "Zunahme", "Verlust", "Masse", "Schätzung">
			misuse
 = <"Nicht zur Dokumentation des ersten Gewichts eines Neugeboren (Geburtsgewicht) - benutzen Sie hierzu den spezialisierten Archetyp OBSERVATION.body_weight-birth.
Nicht zur Dokumentation des angepassten Körpergewichts, z.B. eine Berechnung des vollständigen Körpergewichts bei einer Person mit amputierter Extremität auf Basis der anderen Körperteile und eines Algorithmus - benutzen Sie OBSERVATION.body_weight-adjusted.
Nicht zur Dokumentation eines Objekts oder eines Teils des Körpers.">
			copyright = <"© openEHR Foundation">
		>
		["ru"] = <
			language = <[ISO_639-1::ru]>
			purpose = <"Для записи массы тела человека: фактической или приблизительной.">
			use
 = <"Использовать для записи фактического измерения массы тела, включая случаи отсутсвия части(-ей) тела, врожденное или после хирургического удаления. Отметка о физической неполноте тела может быть зарегистрирована в элементе данных \"стохастическая погрешность\", если требуется. Это - обычный архетип, используемый для типичного измерения веса, например самоизмеренного человеком дома, измерение клинициста в клинике/больнице, или фитнес-инструктором в гимнастическом зале.
```

Figure 3 - Comparison of difference content from archetypes versions

4. CONCLUSION

The usage of online repositories is in constant growth and it can become troublesome to manage all information inside of them. Regarding a clinical knowledge repository based on openEHR artefacts, this content is based on archetypes and templates which internally have very well-defined versioning parameters. Although these versioning parameters can be very useful to distinguish internal changes and variants of artefacts, when a project is being created and the requirements for the clinical information model are defined, archetypes are simply downloaded from the openEHR CKM and saved in the local repository of the on-going project. When new versions for the same archetypes are uploaded to the openEHR CKM, they usually are not updated on the local repository. Normally a local archetype will be updated only when a *bug*, error or new requirement is found during the usage of the software that contains that archetype and a check for new version is made manually on the openEHR CKM.

One of the biggest problems of manually checking versions on openEHR CKM is that it can be really time consuming, for example the archetype comparison mentioned on table 1, took several hours to complete. Usually checking new versions of archetypes is avoided due to the time that it consumes to be finished. Herewith this documentation and the script created is expected to give a quick and reliable archetype verification between the local repositories and the openEHR CKM, in order to make it compliant and updated with the main source and facilitate the managing work of the local clinical knowledge repository owner.

About the script, this is the first version and was developed as a proof of concept (POC) using the archetype HRID to help managing the search in the REST API calls: (*rm_publisher-model_name-rm_class.concept_id.v.major*). This version only checks if the archetype on the local repository has a major version (e.g. v0 to v1) in the openEHR CKM and it returns the uniform resource locator (URL) links from the different repositories allowing to make a side by side comparison of both versions.

In the future it is expected to add the archetype parsing functionality in order to get fully internal versioning differences (*MD5-CAM, build_uid* and *revision*) from the content from either repositories, since these archetypes can also have smaller changes during a period of time.

6. ACKNOWLEDGES

The study was made in collaboration with Faculty of Medicine of Porto University in Portugal and Marand d.o.o in Slovenia under the Erasmus+ mobility programme from April 2018 to July 2018. Also, no potential conflict of interest relevant to this article was reported.

7. REFERENCES

1. OpenEHR Foundation. Who is using openEHR? Deployed solutions in healthcare.https://www.openehr.org/who_is_using_openehr/healthcare_providers_and_authorities [accessed June 2018]
2. Garde, S. (2017). In a Nutshell - What is the Clinical Knowledge Manager? https://openehr.atlassian.net/wiki/spaces/healthmod/pages/2949126/Clinical+Knowledge+Manager [accessed June 2018]
3. Leslie. H. (2017). Authoring, Review and Publication Overview https://openehr.atlassian.net/wiki/spaces/healthmod/pages/2949146/Authoring+Review+and+Publication+Overview [accessed May 2018]
4. Marand ADL Designer modelling web tool. https://ehrscape.marand.si/designerv2/#/designer/login [accessed July 2018]
5. Kalra, D., Beale, T., & Heard, S. (2005). The openEHR foundation. Studies in health technology and informatics, 115, 153-173.
6. OpenEHR Foundation. What is OpenEHR? https://www.openehr.org/what_is_openehr [accessed May 2018]
7. openEHR Specification Program (2018). openEHR Architecture Overview. https://www.openehr.org/releases/BASE/latest/docs/architecture_overview/architecture_overview.html [accessed April 2018]
8. Beale, T. and Heard, S. (2007). An ontology-based model of clinical information. Stud Health Technol Inform.
9. Madsen, M., Leslie, H and E. J. H. S. H. (2010). Sustainable clinical knowledge management: An archetype development life cycle. IOS Press, Volume 151: Health Informatics:115 – 132.
10. Conde, A. M. and Berry, D. (2010). Towards best practice in the archetype development process. Proyecto Fin de Carrera, Trinity College Dublin.
11. OpenEHR Foundation. Clinical Models Documentation and CKM. https://www.openehr.org/programs/clinicalmodels/documentation.php [accessed June 2018]
12. Braga, R. D., de Lucena, F. N., & de Sá Leitão-Júnior, P. (2016). Registro Eletrônico em Saúde Interoperável: os desafios do padrão OpenEHR. Journal of Health Informatics, 8(3).

13. OpenEHR Foundation. Instances of national and other clinical knowledge manager. https://www.openehr.org/programs/clinicalmodels/nationalckms [accessed May 2018]
14. OpenEHR Specification Program (2018), Archetype Object Model 2 (AOM2) Specification.https://www.openehr.org/releases/AM/latest/docs/AOM2/AOM2.html [accessed July 2018]
15. OpenEHR Specification Program (2018), Archetype Object Model 2 (AOM2) Specification - AUTHORED_ARCHETYPE Class discription https://www.openehr.org/releases/AM/latest/docs/AOM2/AOM2.html#_authored_archetype_class [accessed July 2018]
16. OpenEHR Specification Program (2018), Archetype Object Model 2 (AOM2) Specification - ARCHETYPE_HRID Class discription https://www.openehr.org/releases/AM/latest/docs/AOM2/AOM2.html#_archetype_hrid_class [accessed July 2018]
17. Preston-Werner, T. (2018). Semantic versioning 2.0.0 - Meaningful method for incrementing version numbers. https://semver.org/ [accessed May 2018]
18. Grieve, G. (2016). FHIR and openEHR. HL7 FHIR Developer days, Amsterdam.