German Medical Data Sciences 2024
R. Röhrig et al. (Eds.)

doi:10.3233/SHTI240855

Tracking Changes for Inter-Version Interoperability in Heterogeneous Evolving Medical Terminologies

Ralph SCHÄFERMEIER[a,1] , Christoph BEGER[a], Franz MATTHIES[a], Konrad HOEFFNER[a], and Alexandr UCITELI[a]
[a] *Institute for Medical Informatics, Statistics and Epidemiology (IMISE), Leipzig University, Leipzig, Germany*
ORCiD ID: Ralph Schäfermeier https://orcid.org/0000-0002-4349-6726

Abstract. Introduction: Medical terminologies and code systems, which play a vital role in the health domain, are rarely static but undergo changes as knowledge and terminology evolves. This includes addition, deletion and relabeling of terms, and, if terms are organized hierarchically, changing their position. Tracking these changes may become important if one uses multiple versions of the same terminology and interoperability is desired. **Method:** We propose a new method for automatic change tracking between terminology versions. It consists of a declarative import pipeline, which translates source terminologies into a common data model. We then use semantic and lexical change detection algorithms. They produce an ontology-based representation of terminology changes, which can be queried using semantic query languages. **Results:** The method proves accurate in detecting additions, deletions, relocations and renaming of terms. In cases where inter-version term mapping information is provided by the publisher, we were able to highly enhance the ability to differentiate between simple additions/deletions and refinements/consolidation of terms. **Conclusion:** The method proves effective for semi-automatic change handling if term refinements and consolidation are relevant and for automatic change detection if additional mapping information is available.

Keywords. Terminology, Controlled Vocabulary, Data Linkage, Health Information Interoperability

1. Introduction

Standardized medical terminologies and code systems are essential for precise communication among medical professionals and for interoperability between connected electronic health systems. The agreement on global terms and identifiers facilitates collaboration in areas such as medical treatment, diagnosis, medical reporting, documentation, prescription, billing, and scientific research and minimizes errors and misinterpretations.

First attempts to create a standardized categorization of diseases date back to the middle of the 18th century and have over the course of time evolved into what is widely

[1] Corresponding Author: Ralph Schäfermeier, Institute for Medical Informatics, Statistics and Epidemiology (IMISE), Leipzig University, Härtelstraße 16-18, 04107 Leipzig, Germany. Email: ralph.schaefermeier@imise.uni-leipzig.de

known today as the WHO's International Classification of Disease (ICD) [1]. The official first release of the SNOMED Clinical Terms (SNOMED CT) system was in 2002 with its origins dating back as far as to 1965 when its predecessor, the Systematized Nomenclature of Pathology (SNOP), was incepted [2].

While medical terminologies vary in intended application and covered domain, they share certain communalities about their structure. They all provide means for identifying and naming the relevant concepts of a particular domain of interest and for describing properties of and relations between them. The standard terminological primitives used for this purpose are *codes, terms, synonyms,* and *definitions*.

The features of terminologies allow to establish a static representation of knowledge about relevant aspects of a domain. However, medical knowledge is seldom static but evolves over time as new phenomena are observed, new mutations of pathogens and diseases emerge, new clinical drugs are discovered, and new diagnostic and treatment procedures are developed. Once established knowledge may become obsolete, or existing structures may be reorganized. To remain accurate representations of the real world, terminologies are required to evolve along with the knowledge that they reflect. Consequently, terminologies are usually constantly updated, and new versions are released in more or less regular release cycles.

This evolutionary aspect is common to most terminologies, whereas the release mechanisms and their respective outcomes vary significantly between publication sources. Some terminologies include release notes with change documentation, while others do not. If available, the change documentation is often not in a structured, machine-readable format but human-readable text.

Changes may include the addition of new terms, deletion or deprecation of obsolete terms, adding or removing of synonyms, correcting spelling errors, adding international labels, and rearranging terms within the terminological hierarchy. For use cases such as coding for the purpose of billing or documentation, this might not be problematic if it is made clear which terminology version a used code or term refers to. In other scenarios however, interoperability between different versions of the same terminological system is desirable. An example for such a scenario is the selection of features from distributed, heterogeneous medical databases for the purpose of conducting scientific studies or cohort selection for clinical trials. The data in such systems have potentially been collected over a long period of time, and different versions of terminologies might have been used for labeling individual entries, which leads to the risk of queries missing relevant records or the retrieval of irrelevant records containing terms the meanings of which have changed over time. Consequently, parties using these terminologies in computer systems and relying on interoperability between versions face the necessity to perform a considerable amount of manual adaptation work, especially when several terminologies (possibly with differing update cycles) are used in parallel.

In this paper, we propose a novel approach for the unified use and updating of heterogeneous, evolving medical terminologies. The method consists in a data processing pipeline that allows the import of multiple terminologies in different source formats. Semantic as well as lexical differences between versions are detected, and the information about the differences are stored alongside the terminologies themselves. The combined terminological and change information allows for an automated recommendation of inter-version mappings of terms, while presenting users with the complete change history for each term. The aim is the preservation of the lexical and semantic meaning of terms throughout the version history.

The proposed method is a partial result of the project Terminology and Ontology-based Phenotyping (TOP), in the context of which multiple terminologies from heterogeneous sources (and all with heterogeneous release mechanisms) are used for describing and searching for simple and complex phenotypes in distributed external sources [3].

2. Related Work

The first formal account of change operations in medical terminologies was given in [4]. The authors provide a sound and complete survey of fundamental change operations as part of an overall conceptual model of terminologies. The paper remains on an abstract and descriptive level and does not treat the technical aspects of the problem of change *detection*. The conceptual model constitutes, however, the basis of this work.

Later works address the problem of changes in more expressive (and more general) representation formalisms, such as description-logic-based ontologies [5, 6]. They focus on the semantic or logical consequences of atomic changes (as the semantics of these formalisms are usually defined in terms of some subset of first order logic).

The authors of [7] provide a formal account for changes in RDF graphs, considering both semantic and lexical changes. The focus of this work lies, however, on the problem of consistency preserving changes and proposes evolution patterns satisfying that requirement.

In [8], the author proposes a versioning model for (medical and non-medical) terminologies. However, the work focuses on describing term versions (in terms of timestamps) and not the changes involved in a transition between versions.

To our knowledge, none of the prior works in this area have addressed the problem of automated change tracking in (medical) terminologies for the purpose of establishing interoperability between different versions of the same terminology.

3. Methods

The proposed method consists of a data transformation pipeline, involving several harmonization, analysis, and retrieval steps.

The first step consists in the import of existing terminologies and their transformation into a common format to establish a common ground for semantic interoperability. This comprises the definition of a data mapping from the various possible input formats to our common representation format. We specify the common format in the form of an ontology, the *TOP Terminology Ontology* that captures the relevant concepts and relations of a terminology, such as *term, identifier, code, synonym, parent term* and *child term*. Sections 3.1 and 3.2 contain a detailed description of this step, its underlying rationale, and the structure of the ontology.

In the second step, changes between different versions of the same terminology are detected. Since terminologies encode lexical and semantic (such as hierarchical) information, all these types of changes are relevant and need to be detected. The detected changes for each pair of subsequently released versions of the same terminological system are stored. A second ontology of change types that are relevant in the context of terminological systems, the *TOP Change Ontology,* has been developed for this purpose. Section 3.3 describes this step in more detail.

The resulting terminological and change data are then stored and made available via query endpoints that can be used by an application (in our case, our TOP framework) for calculating possible update actions between older and newer terms that are used in the application.

Figure 1 shows the architecture of a system implementing the pipeline, with each component representing one of the steps.

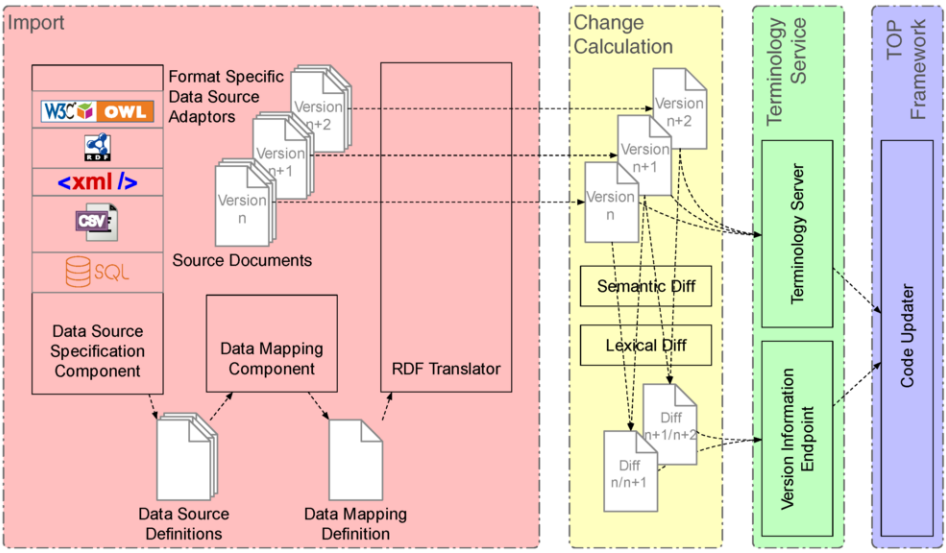

Figure 1. The system architecture.

3.1. Data Source Mapping

Available terminological sources vary in three aspects: their serialization format, their representation model, and their underlying data schema. Harmonization of heterogenous sources therefore requires taking into consideration these three aspects.

The *serialization format* determines on a technical level how data is stored on a medium. It defines rules for encoding an abstract representation of data in terms of character sequences and for decoding character sequences back into the abstract representation. Examples of serialization formats include Comma-Separated Values (CSV), the Extensible Markup Language (XML), the JavaScript Object Notation (JSON), and YAML. There may be subtypes (often referred to as dialects) of serialization formats. For example, different CSV files may use different characters for the separation of data units, such as comma, semicolon, or colon.

The *representation model* defines, on an abstract level, semantic primitives for describing data and relations between data. Many standards do not make a clear distinction between their representation model and their serialization format. An example is XML, which prescribes a one-to-one correspondence between the primitives of the abstract representation model (such as XML elements) and their serialization (XML tags). An example of a standard that makes a clear distinction between the two is the Resource Description Framework (RDF), which defines its representation model in terms of graph structures of interconnected resources and caters for a broad and constantly increasing variety of serialization formats.

The representation model also determines the *data access paradigm*. While, for example, CSV files and relational databases may have completely different serializations, they share the same abstract representation model (which consists of tables with rows and columns) and thereby allows for a common data access paradigm in the form of SQL queries. JSON and XML also share a common data access paradigm (XPath), while RDF's primary access paradigm is graph queries (for example using the graph query language SPARQL).

A comprehensive mapping approach needs to encompass all of the three aspects as well as the conceptual blur that exists in such cases as XML or JSON. Furthermore, input sources may be distributed across multiple files (possibly having different formats). An example is the German version of the ICD-10 vocabulary, which is split into two CSV files: one for the ICD-10 codes and categories, and one for the lexical labels (with each file having a different separator character). Handling multi-file input sources is thus a further requirement for the system. In the context of this work, input sources in the formats CSV (with varying separator characters), RDF (in n3 and rdf/xml serializations), and XML, in single and multi-file configurations have been encountered.

3.2. Translation

Once a unified access to the various source types has been established, the translation to a common output format is straightforward. In our implementation, we have used the eclipse BIRT reporting engine and implemented a custom emitter that performs the transformation. The result of the translation of each source file is an RDF file adhering the schema specified by the TOP Terminology Ontology.

Figure 2 shows an example of a snippet of data conforming to the ontology.

3.3. Change Calculation

Change calculation between two subsequent versions of a terminology involves the pairwise comparison of relevant primitives encountered in the unified presentation. As mentioned in Section 1, lexical and semantic changes are relevant.

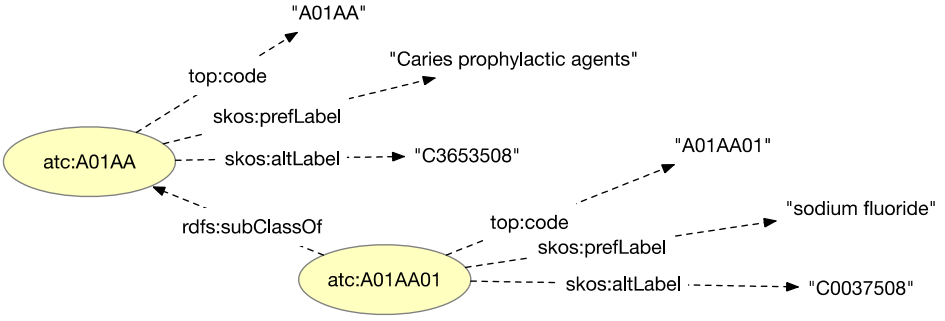

Figure 2. An example of a data snippet conforming to the TOP Terminology Ontology. The yellow ovals represent terms from the Anatomical Therapeutic Chemical (ATC) classification system. The terms are arranged hierarchically (specified by the `rdfs:subClassOf` relation). Identifiers are represented using the `top:code` relation, and preferred label and lexical synonyms using the properties `skos:prefLabel` and `skos:altLabel` from the SKOS vocabulary. Besides lexical synonymity semantic synonymity between terms can also be expressed using the `owl:equivalentClass` property from the OWL vocabulary.

3.3.1. *Semantic Changes*

Semantic changes may comprise the introduction or deletion of a term, changes in the hierarchical position of a term, and changes of semantic synonym relations between two or more terms. The latter two involve addition and/or deletion of either hierarchical or non-hierarchical connections. Detecting changes in these relations therefore involves the pairwise comparison of corresponding statements in each of the two RDF graphs.

Detection of location changes of a term (or its entire subbranch) is accomplished by comparison of the sets of `rdfs:subClassOf` axioms that are associated with the older and newer version of the term, respectively. Most terminologies (at least the ones treated in the context of this study) have non-faceted hierarchical structures, i.e., each term (with the exception of the root term) has exactly one super term. The detection of the relocation of a term to a different super term therefore involves the detection of the deletion of a subclass axiom and the addition of a new subclass axiom.

3.3.2. *Lexical Changes*

Possible lexical changes include the addition and deletion of labels, renaming (which involves the deletion of the label and the addition of a new one using the same property), and the switching of the role of preferred label and synonym.

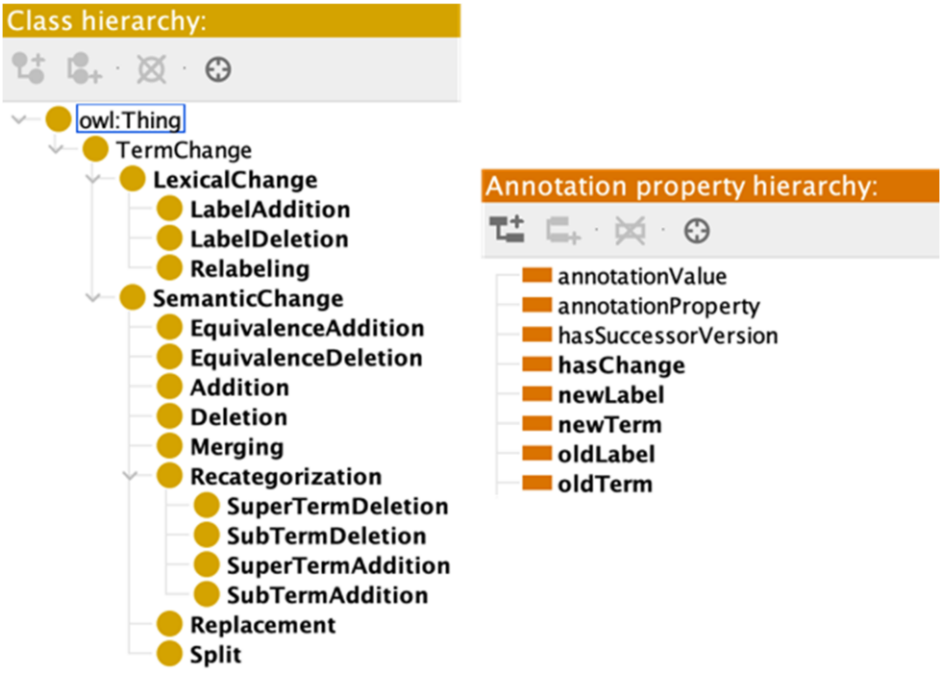

Figure 3. The TOP Change Ontology

3.3.3. *Representation of Changes*

Once detected, changes between two versions of a terminology are stored in an RDF graph, adhering to the schema prescribed by the TOP Change Ontology (see Figure 3).

Figure 4 shows an excerpt of a change graph for a comparison of two versions of the ATC terminology. The two versions under comparison are specified in the default graph. All changes reside in a named graph (the graph IRI of which is the IRI of the newer ontology version that incorporates the changes). The example shows the representation of the addition of the label "ferrous amino acid complex and folic acid" to the term "B03AD01". Changes are connected to the affected term using the `hasChange` property. Changes are represented as anonymous (blank) nodes. Their respective type is represented by one of the classes from the TOP Change Ontology. Depending on the change type (lexical or semantic; addition, deletion, or replacement), the appropriate relation types are used for either specifying the label or the relational partner term, if applicable. Pure term additions and deletions do not have any further properties.

```
{
  atc:2023aa v:hasSuccessorVersion atc:2023ab .
}

atc:2023ab {
  atc:B03AD01 v:hasChange _:30a8cce2f9a34dd9952fd7f3af72ef262619,
                          _:30a8cce2f9a34dd9952fd7f3af72ef262620,
                          _:30a8cce2f9a34dd9952fd7f3af72ef262621,
                          _:30a8cce2f9a34dd9952fd7f3af72ef262622 .

  _:30a8cce2f9a34dd9952fd7f3af72ef262620 a v:LabelAddition;
    v:annotationProperty skos:prefLabel;
    v:annotationValue "ferrous amino acid complex and folic acid"^^rdf:PlainLiteral .
}
```

Figure 4. An example of the representation of a lexical change in trig notation. The first line (between curly braces) is in the default graph and describes the fact that ATC release 2023ab is the successor version of the ATC release 2023aa. The subsequent lines are statements belonging to graph arc:2023ab, which contains all change information for that particular version. In this excerpt, one can see that the term with ATC code B03AD01 has four changes (each given a generated id). Change 30a8cce2f9a34dd9952fd7f3af72ef262620 is a label addition (ferrous amino acid complex and folic acid).

This representation format allows for efficient querying using SPARQL queries. If only changes between a particular pair of terminology versions are of interest, one can specify the corresponding named graph in the query. On the other hand, it is possible to formulate cross-graph queries if one is interested in the change history of a particular term across multiple version changes.

3.3.4. Implementation Details

The import and translation pipelines were implemented on the basis of the Eclipse Data Tools Platform[2] and the BIRT reporting engine[3], which provides graphical user interface components for data access and mapping definitions from the different input terminology formats to or common data model. A custom emitter has been implemented that transforms the various input sources to an OWL ontology.

For the detection of the semantic changes, we use an extended version of the ontology diff tool bubastis[4], which was further extended to also detect the lexical changes and output the representation format described in Section 3.3.3. The semantic extensions include consolidating the deletion and addition of subclass axioms to a hierarchy change and detecting splitting and merging of terms were possible.

[2] https://projects.eclipse.org/projects/tools.datatools
[3] https://projects.eclipse.org/projects/technology.birt
[4] https://github.com/Onto-Med/bubastis-semlex

4. Results

The proposed system was implemented as described in Section 3 following the architecture shown in Figure 1 and tested with seven medical terminologies (ASK (the German catalog of medicinal products), ATC, the German version of ICD-10, LOINC, OPS, PZN (the German pharmaceutical drug catalog), and SNOMED CT. The number of available versions ranged from 2 (ASK) to 18 (ATC).

In order to assess the accuracy of the approach, we used mapping information that was provided along with ICD10GM and OPS code systems by their publisher, the German Federal Institute for Drugs and Medical Devices. We analyzed 14 releases in total, seven of each ICD10 and OPS, spanning the years 2016-2022, respectively. As the code systems and their inter-version mappings are large (the OPS 2022 release contains over 37,000 codes with more than 45,000 labels and over 33,000 inter-version mappings), manual processing proved unfeasible, and we decided for a programmatic approach. The code extracting the change information from the mapping files is publicly available[5].

The change information is provided in a CSV format and contains one-to-one mappings for each code. In case of a code split, the code appears one time in the old revision column and several times in the new revision column, and vice versa in case of a code merge. We used this information to infer merge and split operations following the rule that merging arbitrarily many terms in the older version into one term in the newer version counts as one merge operation. Vice versa, we counted the split of one term into arbitrarily many new terms as one operation.

As our approach is not able to reliably predict merges and splits due to the lack of one-to-one mapping information (if new terms are added with the intention to refine an existing term, without the mapping information, it is not possible to infer that intention), the results where highly skewed in the first run. The average numbers of label additions and deletions detected by our approach without external change information were 943% and 1515% of those detected by the evaluation approach using external change information. This is explainable by the fact that our approach without external change information counts merges (which occurred 325 times on average per new revision) and splits (212 times on average) as all the atomic add/delete operations they consist of.

Using the external change information in our approach we were able to achieve 100% accuracy in for split and merge operations.

5. Discussion

As described in Section 4, the accuracy of the approach can be highly improved if external term mapping information is available, especially for distinguishing between simple additions and deletions of terms on the one hand and merge and split operations on the other hand.

Even without the ability to distinguish these operations, the approach still proves useful in scenarios, such as phenotype definition and cohort assembly.
During the conception of the presented system a series of representational choices had to be made.

[5] https://github.com/Onto-Med/gmds2024-evaluation

In the TOP Change Ontology, for example, the choice was made to represent terms as OWL classes. The underlying rationale is based on the fact that terms are represented in the same way as in Terminology Servers, which are used as the storage and retrieval component for terms in our system. A different way for representing terms would have been in the form of OWL individuals. The latter would have had the benefit of being able to use object property relations between terms (whereas classes can only have annotation properties) and being able to use the more expressive semantics of OWL. On the other hand, the representational power of RDF in combination with SPARQL has proven to be sufficient for the intended purpose of reconstructing semantic and lexical change histories of terms.

A further choice that has been made is the representation of versions in terms of RDF named graphs. Alternatively, a change could have been represented by a version annotation on each change instance. Regarding the intended purpose, both variants are equally expressive, and the preference for named graphs was merely driven by performance considerations (restricting the selection to a named graph requires one join operation less than resolving the associated version via a dedicated annotation property).

One open problem that was not solved in the context of this work is the distinction between two unrelated add and delete operations and a replacement (or, in the lexical case, renaming) operation. The problem is not solvable without additional background information. However, certain cases may be resolved using special heuristics. Certain types of lexical renaming (especially spelling error corrections) could be determined using lexical similarity metrics, such as the Levenshtein distance.

The modeling of singular changes (addition and deletion of terms) does not require the creation of a dedicated blank node, as the change does not have further properties other than the associated term and its type. The choice to use blank nodes in these cases nevertheless was driven by the preference for a uniform representation schema and avoiding unnecessarily complex SPARQL queries. Furthermore, it is conceivable that later versions of the system produce additional information, which might be useful to the end user. In that case, a distinct node will become necessary.

6. Conclusion

We have presented a method for tracking and representing changes of terms in heterogeneous, evolving medical terminologies. Future work will consist in the incorporation of further terminologies, possibly requiring an extension or modification of the representation model. A further open problem that will be interesting to address in the context of future work is the treatment of mappings between terminologies. So far, we have only investigated changes in individual terminologies. However, in an effort to make individual terminologies interoperable with another, some providers publish additional term mappings between individual pairs of terminologies. These mappings need to evolve alongside the terminologies they connect, and it would certainly be beneficial to include them in our approach.

Declarations

Conflict of Interest: The authors declare that there is no conflict of interest.

Contribution of the Authors: RS: conception & implementation of the method, conduct of the experiments and writing of the manuscript; CB, FM, CH, AU: discussion contributions to the conception and proof-reading of the manuscript. All authors have approved the manuscript as submitted and accept responsibility for the scientific integrity of the work.

Funding: The SMITH consortium was funded by the German Federal Ministry of Education and Research (grant number: 01ZZ1803A), the 'Terminology and Ontology-Based Phenotyping (TOP) project is funded by the German Federal Ministry of Education and Research (grant number: 01ZZ2018)

Implementation: https://github.com/OntoMed/terminology-transformationspecs

References

[1] Hirsch JA, Nicola G, McGinty G, Liu RW, Barr RM, Chittle MD, et al. ICD-10: History and Context. AJNR: American Journal of Neuroradiology. 2016 Apr;37(4):596-9. doi:10.3174/ajnr.A4696.

[2] Gaudet-Blavignac C, Foufi V, Bjelogrlic M, Lovis C. Use of the Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT) for Processing Free Text in Health Care: Systematic Scoping Review. Journal of Medical Internet Research. 2021 Jan;23(1):e24594. doi:10.2196/24594.

[3] Beger C, Matthies F, Schäfermeier R, Kirsten T, Herre H, Uciteli A. Towards an Ontology-Based Phenotypic Query Model. Applied Sciences. 2022 Jan;12(10):5214. doi:10.3390/app12105214.

[4] Oliver DE, Shahar Y, Shortliffe EH, Musen MA. Representation of Change in Controlled Medical Terminologies. Artificial Intelligence in Medicine. 1999 Jan;15(1):53-76. doi:10.1016/S0933-3657(98)00045-1.

[5] Klein M, Fensel D, Kiryakov A, Ognyanov D. Ontology Versioning and Change Detection on the Web. In: Gómez-Pérez A, Benjamins VR, editors. Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web. Berlin, Heidelberg: Springer; 2002. p. 197-212. doi:10.1007/3-540-45810-7 20.

[6] Jonquet C, Poveda-Villalón M. About Versioning Ontologies or Any Digital Objects with Clear Semantics. In: DaMaLOS 2023 - 3rd Workshop on Metadata and Research (Objects) Management for Linked Open Science. Hersonissos (Crète), Greece: L. J. Castro and S. Schimmler and J. Dierkes and D. Dessì and D. Rebholz-Schuhmann; 2023. doi:10.4126/FRL01-006444994.

[7] Auer S, Herre H. A Versioning and Evolution Framework for RDF Knowledge Bases. In: Virbitskaite I, Voronkov A, editors. Perspectives of Systems Informatics. Berlin, Heidelberg: Springer; 2007. p. 55-69. doi:10.1007/978-3-540-70881-0 8.

[8] Schnieder L. Formalisierte Terminologien Technischer Systeme Und Ihrer Zuverlässigkeit [Berichtsreihe]. DLR-Institut für Verkehrssystemtechnik; 2010.