



Mapping of Change Operations from Gene Ontology into Medical Subject Headings

Anny Kartika Sari^{1*}

¹*Department of Computer Science and Electronics, Universitas Gadjah Mada, Indonesia*

* Corresponding author's Email: a_kartikasari@ugm.ac.id

Abstract: Concept mapping between two different ontologies in the biomedical field are commonly conducted. When one of the ontologies changes, the changes should be propagated to the other ontology in the mapping. This propagation process is an issue that must be addressed because the knowledge in biomedical field develops rapidly. In this paper, we present the change propagation method from Gene Ontology (GO) to Medical Subject Headings (MeSH). The change propagation is conducted by mapping GO change operations into MeSH change operations. There are three steps that must be performed. Firstly, the change operations on each ontology are identified and represented formally. Secondly, change operations in both ontologies are classified and compared. Finally, each GO change operation is mapped into a MeSH change operation based on the semantic meaning of the change operation. A proof of concept is also presented to show how the process works. In the experiment, the method has been applied to ten concept deletion operations of GO, with 80% of operations being successfully propagated to MeSH. The 100% success cannot be achieved because some GO concepts do not have relevant corresponding MeSH concepts due to the specificity of GO concepts.

Keywords: Change propagation, Gene ontology, Mapping, Medical subject headings, Ontology evolution.

1. Introduction

An ontology is a formal representation of knowledge in a specified domain of interest. Ontologies in various fields are available today. Each of them is shared among applications in a specific field to support the semantic interoperability between these applications.

Biomedicine is one of the fields in which ontologies are used intensively. In biomedical field, ontologies are usually referred to as terminologies or vocabularies. In this paper, the term ontology and terminology are used interchangeably.

There are several standardized terminologies in biomedical fields such as SNOMED CT (Systematized Nomenclature of Medicine -- Clinical Terms), MeSH (Medical Subject Headings), LOINC (Logical Observation Identifier Names and Codes) and GO (Gene Ontology). Each terminology has its own function and use, and is governed by a specific institution that manages, updates and distributes the

terminology. SNOMED CT is the most comprehensive clinical healthcare terminology in the world [1], while MeSH is a controlled thesaurus for indexing, cataloguing, and searching of biomedical and health-related terms [2]. LOINC contains a set of identifiers, names, and codes for identifying health measurements, observations, and documents [1], while GO provides a computational representation about the functions of genes [3].

Most medical terminologies are integrated into a large vocabulary that is called UMLS (Unified Medical Language System), more specifically UMLS Metathesaurus. It contains concepts related to biomedicine and health, their names and relationships among them. UMLS is developed by the US National Library of Medicine (NLM).

Other than UMLS, MeSH is also managed by NLM. MeSH has been used for various purposes. First, MeSH can be used as a medium to achieve semantic interoperability in biomedical and health data. For example, the terms used in health

archetypes [4] can refer to concepts that exist in MeSH and other terminologies in biomedical field. This is to achieve semantic interoperability among applications in different health institutions. The concepts in MeSH can be used both as key words and annotations to be embedded in each document. Keywords and annotations are beneficial for information retrieval purpose. At this time, concepts in MeSH have been used to annotate articles in biomedical related domain such as pharmacies [5]. The aim of the annotations is to give general description about the content of the document, and sometimes is used for indexing as well.

Besides MeSH, Gene Ontology (GO) is also a standardized ontology that is widely used in the biomedical field. GO is managed by the Gene Ontology Consortium. GO has been used for various purposes, including as a universal standard for identifying gene functions so that there is semantic uniformity in these functions. In [6], GO can be used as a basis for predicting gene function, in which the GO-based semantic Hierarchy Preserving Hashing (HPHash) method was developed to predict gene function. Furthermore, in [7], GO is used to annotate datasets, from which the weighted association rules are then extracted. In addition to these very specific uses, the very widespread use of GO occurs in the field of molecular biology.

It is common that a concept in an ontology is mapped into a concept in another ontology in the biomedical field. One example is the mapping of concepts in SNOMED CT into concepts in ICD-10 and ICD-10-CM [8]. In another example, application like archetype [4] also contains the mapping of its terms to more than one ontologies. Implicit mapping is also applied in UMLS, in which a specific UMLS concept can be obtained from different source ontologies. This means that actually the concepts from different source ontologies represent the same meaning, hence, only one UMLS concept represents them.

Mapping of concepts can also be conducted from GO to MeSH. The mapping can be done from GO to MeSH and vice versa. However, since GO is a specific ontology that contains concepts related to genes while MeSH contains concepts related to more general biomedical and health field, the reference from MeSH to GO is more likely to happen. As MeSH contains more general knowledge, the number of concepts that are related to gene functions is not large. Hence, only a subset of concepts in MeSH can refer to concepts in GO. The reference can be seen as the approach to get the precise meaning of the concepts from the ontology that is intended to represent the knowledge related to gene functions. In

UMLS Metathesaurus, such reference has been identified implicitly between vocabularies, including MeSH and GO. For instance, the term *Reproduction with CIU* (Concept Unique Identifiers) C0035150 represents a GO concept (GO:0000003) as well as a MeSH concept (D012098), among other concepts in other vocabularies.

The knowledge in biomedical field develops quickly. This is supported by the rapid progress on biomedical research. When knowledge changes, the underlying ontology must be adjusted to accommodate the changes. In ontological term, this process is referred to as an ontology evolution. These changes will affect all applications that refer to ontological concepts. For example, annotations on documents based on MeSH concepts must be adjusted so that the annotations always refer to the concepts that exist in the latest version of MeSH. In addition, changes will also affect the mapping of concepts between ontologies. Mapping must be kept updated so that it represents the current state of each ontology.

One way to maintain the currency of the mapping is to do the mapping of change operations from one ontology to another ontology. In this way, when a concept in a referred (target) ontology changes, the change can be propagated to the other ontology (source ontology) that refers to the target ontology. The propagation can be conducted by initiating the corresponding change operation. The initiation of change operation indicates that a certain change should be performed in the source ontology.

This research focuses on mapping the change operations between different ontologies, specifically between GO and MeSH. This issue has not been addressed by the existing research, which usually focuses on the evolution of single ontology. Several steps must be conducted to map change operations. The first step is to classify the change operations in each ontology. The second step is to represent formally the change operations in each ontology. The next step is to compare the characteristics of each change operation between different ontologies. The final step is to devise an algorithm to map change operations from one ontology to the other.

The rest of the paper is organized as follows. Section 2 presents related work, followed by section 3 that presents preliminaries on GO and MeSH. Section 4 discusses research methodology, while section 5 addresses the formal representation and classification of change operations. In section 6, mapping of change operations from GO to MeSH is

Table 1. Ontology evolution phase addressed by the latest existing work

Research	Ontology Evolution Phase		
	Representation	Manipulation	Propagation
[11]	x		
[12]	x		
[13]	x		
[14]	x		
[15]	x		
[16]		x	
[17]		x	
[18]			x
[19]			x
[20]			x

explained, followed by discussion in section 7. Section 8 concludes this paper.

2. Related work

Different phases of ontology evolution has been defined in [9-10]. First phase division of ontology evolution is proposed in [9], in which an ontology evolution is divided into six phases, namely change capturing, change representation, semantics of change, change implementation, change propagation and change validation. In [10], the number of phases is reduced to three, i.e. ontology ontology change representation, ontology change manipulation, and ontology change propagation. Even though each of both works has its own phases, the phases are comparable. In this related work discussion, we use the phases proposed by [10].

The change representation phase of ontology evolution has been discussed in [11-15]. In [11], Pi-Calculus is used to model and represent the semantics of change operations. The operational semantics of change are formalized as a series of information exchanges from these agents. Similarly, a study in [12] is conducted to formally identify and represent change operations at LOINC, which is completed with the algorithms for the identification of these operations. In [13], an ontology that represents changes of language is built, in which the changes are represented as instances of the ontology. The work in [14] is more related to change capturing phase. In this work, the information from DBpedia is extracted as the source of evolution. Then, the change operation is defined based on the information. The work in [15] uses Historical Knowledge Graph (HKG) to represent the changes between different versions of ontology.

Change manipulation phase has been the focus of the work in [16-17]. A method of automatic ontology adjustment has been proposed in [16] to represent dynamic nature of the manufacturing data environment. This method is a part of the whole

framework proposed in this work, with the objective of implementing intelligent manufacturing application. Ontology maintenance becomes the focus of [17]. In this work, a method is proposed to keep the ontology updated. The work is applied to the ontology representing the knowledge related to Rheumatoid disease.

Change propagation phase is also an interesting topic to be discussed. In [18], the propagation of changes in ontology to the established concept mappings is addressed. The neighbourhood of aligned concepts is explored to update the mapping sets. The work in [19] also addresses the refinement of mappings due to ontology evolution. This work uses LOINC releases in different languages to evaluate the proposed method. In [20], dynamic ontology is used to describe the semantics of devices incorporating in the Web of Things and enable the dynamic interactions between devices.

Table 1 shows the summary of the recent existing work related to ontology evolution, specifically showing the phase of ontology evolution that is addressed in each work. It is shown that each recent work related to ontology evolution addresses only one phase of ontology evolution.

Different from existing work presented in Table 1, in this work, two phases of ontology evolution are addressed, i.e. ontology change representation and ontology change propagation. In this case, the formal representation of change operations in GO and MeSH is presented. Moreover, the change propagation is addressed between two biomedical related ontologies. This kind of propagation has not been addressed by existing work. The change propagation in this case is from GO to MeSH. The goal of the propagation is to identify possible changes that are needed in MeSH based on the changes in GO.

We also have conducted literature review to existing works related to MeSH. Most of them discuss MeSH with regard to the use of MeSH terms for annotation. In [21], an evaluation has been

conducted to examine the use of MeSH terms to improve search effectiveness for different kinds of users. In [22], MeSH terms are used to identify publications within specific categories as the knowledge representations of demand, supply, and technological capabilities. In [23], MeSH is one of several metathesaurus that are compared in terms of descriptive terms, structure, and features of Traditional Chinese Medicine interventions. Research conducted in [24] addresses the evolution in MeSH, in which a temporal classifier is established to predict which parts of MeSH will develop. Error analysis of queries for searching documents on MEDLINE is conducted in [25]. The focus of the research is on the MeSH terms contained in the query and the difference between the potential recall and actual recall.

Like research related to MeSH, research on GO generally discusses the use of GO. As previously mentioned, GO is also used as a basis in research to predict the function of the gene [6]. The research introduces a semantic method based on HPHash (Gene Ontology Hierarchy Preserving Hashing) for the prediction of gene function. Hash techniques are also used in [26], in which Hashing GO (HashGO) method is proposed for protein function prediction to accurately predict the association between the GO term and a very large number of proteins. Furthermore, in [27], sixteen different semantic gene-gene dissimilarity measures are proposed through clustering based on GO terms. Similarly, in [28], gene ontology (GO) annotations are proposed based on semi-supervised clustering algorithm called GO fuzzy relational clustering (GO-FRC).

The significance of our proposed work relies on two issues. Firstly, we address two phases of ontology change evolution, i.e. change representation and change propagation. Secondly, the change propagation process is conducted between two biomedical ontologies, i.e. Gene Ontology and MeSH. Using the method, if a GO concept is changed, the change can be propagated into the corresponding MeSH concept, if available. This addresses the issue of difficulty in maintaining the mapping between two different ontologies due to the changes happen in one of the ontology.

3. Preliminaries on GO and MeSH

In this section, the structures of GO and MeSH are described briefly. The structures will determine how GO and MeSH, including the change operations in both ontologies, will be represented formally.

3.1 GO structure

GO consists of three ontologies, namely Cellular Component Ontology, Molecular Function Ontology and Biological Process Ontology. GO can be represented as directed graph, but it is not a tree. There are more than 40,000 concepts in GO.

GO essential term elements are unique identifier and term name, aspects, definition and relationship to other terms. The unique identifier consists of 7 digit identifier (also called term accession or term accession number), which is preceded by GO:. Some examples are GO:0005125 and GO:0060092. Term name is human-readable name for a term such as *mitochondrion*, *glucose transport*. Definition is a description of the terms and a reference to the source of information. Relationships to other terms is the link that connects one term to another. The main relationship is *is a*. An example of relationship is GO: 0015758: *glucose transport* is a GO: 0015749: *monosaccharide transport*. Other types of relationship that are widely used in the ontology are *part of*, *has part* and *regulates*. There are additional elements like Secondary IDs (Alternate ID), Synonyms, Database cross-references, Comment, Subset and Obsolete tag.

Obsolete tag is very related to the information about ontology evolution. It indicates outdated terms that should not be used. This occurs to a term that is out of scope, named or incorrectly defined. In this case, Term and Term ID are still in the ontology, however, the term is tagged as obsolete.

3.2 MeSH structure

MeSH can be viewed based on two points of view. The first viewpoint is that MeSH consists of a collection of records and the relationship between these records. In this case, the record refers to the subject, which can be used to index articles. In the second point of view, MeSH is a collection of Concepts. Concept is a collection of Term on MeSH records that are synonymous with one another. Term is a meaning. Thus, a record consists of several Concepts, while each Concept consists of several Terms.

In this study, the used viewpoint is that which views MeSH as a collection of records. This point of view is more suitable for discussion because in reality, MeSH is more widely used to annotate documents. The record is used to do the annotation. Basically, there are three basic types of records from MeSH, namely Descriptors, Qualifiers, and Supplementary Concept Records (SCRs). In this paper, only Descriptors and Qualifiers are considered because

SCRs are basically the supplement of the other two types of records. Each SRC is only linked to one or more Descriptors. No hierarchy is formed by the SCRs.

Descriptors are used to index citations in the MEDLINE database and to catalog publications. Descriptors are also called Main Headings. In ontological term, Descriptor can be considered as concept. Each Descriptor has MeSH heading name, Unique ID and Tree Number(s) among other things. An example of Descriptor is *Glucose* with Unique ID of *D005947* and Tree Number of *D09.947.875.359.448*. There are 16 categories of Descriptors. Each category is divided into subcategories, of which is again divided into more specific descriptor. This forms a tree of up to thirteen hierarchical levels. Hence, the relationship type that is used mostly in the tree hierarchy is *IS-A*, where a Descriptor is a specialization of a more general there are several other types of relationships that indicate the relationship between Descriptors such as *SeeRelatedDescriptor*, which is used to remind users of a Descriptor that other Descriptors may be more suitable for use in a particular context, and *ConsiderAlso*, which is used to refer to a group of Descriptors that have the same linguistic root.

Qualifiers are used together with Descriptor to index and catalog items. Qualifier is also referred to as Subheading. At present, there are 83 Qualifiers. Qualifiers are usually used to group items that are indexed according to aspects of a subject. For example, indexing *Liver / drug effects* (in this case, *Liver* is Descriptor, while *drug effects* is Qualifier) indicates that the article addresses the effects or effects of drugs on Liver, and is not an article about Liver in general. In MeSH the types of Qualifiers for each Descriptor have been specified. Conversely, for each Qualifier, Descriptors that can use it also have been determined. Therefore, MeSH users are not allowed to choose Descriptor of their choices if the pairing is not specified in MeSH.

4. Research methodology

For interoperability purposes, concepts in GO can be mapped into concepts in MeSH, and vice versa. However, as previously stated, since GO contains more specific domain than MeSH, the direction of the mapping tends to be from MeSH into GO.

Fig. 1 describes the phases to do the mapping from change operations of GO into change operations of MeSH. Detailed explanation about each phase is as follows.

1. Identifying change operations in GO and MeSH.

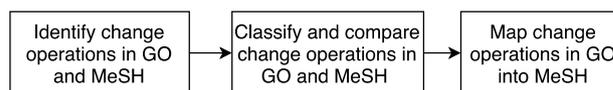


Figure. 1 Phases to do the mapping of change operations

In this phase, change operations in each ontology is identified. The releases of GO and MeSH must be examined to find differences between two releases to determine types of change operations. The change operations are then represented formally.

2. Classification and comparison of change operations in GO and MeSH.

In this phase, the change operations in each ontology is classified. Then, the change operations in both ontologies are compared to identify the characteristic similarities between them. This is to achieve the correct mapping of change operations.

3. Mapping of change operations from GO into MeSH.

The mapping is based on the similarities between change operations in both ontologies. Here, a change operation found in GO is then propagated to MeSH. This is done by finding a concept in MeSH, through UMLS Metathesaurus, that represents the similar meaning with the changed concept in GO. UMLS is used because it contains both GO and MeSH. A concept in UMLS represents a meaning, and can be obtained from more than one ontologies. Based on this fact, the rule is as follows: if a GO concept changes, and the UMLS concept that represents it also represents a MeSH concept, then the change of GO concept should be propagated to the MeSH concept.

5. Formal representation and classification of change operations

In this section, change operations in GO and MeSH will be explained. For each ontology, formal representation will be presented. Then, the change operations are identified. For better understanding of the formal representation, the notations used in the formalization is listed in Table 2.

5.1 Change operations in GO

Two basic elements in GO are terms and relationships. A term in GO is basically a concept in an ontology, while a relationship is also a relationship in an ontology. Hence, the formal definition of GO ontology is shown by Definition 1.

Table 2. List of notations

Notation	Definition
O_X	Ontology X
C_X	Set of concepts in ontology X
R_X	Set of relationships in ontology X
Q_M	Set of MeSH Qualifiers
M_M	Set of MeSH mappings connecting Descriptors and Qualifiers
c_{new}	Concept to be added to the ontology
c_{del}	Concept to be deleted from the ontology
r_{new}	Relationship to be added to the ontology
r_{del}	Relationship to be deleted from the ontology
q_{new}	Qualifier to be added to the MeSH
q_{del}	Qualifier to be deleted from the MeSH
m_{new}	Mapping to be added to the MeSH
m_{del}	Mapping to be deleted from the MeSH

Definition 1: GO Ontology

$O_{GO} \equiv \langle C_{GO}, R_{GO} \rangle$ is GO ontology with:

- C_{GO} is the set of GO concepts, referring to GO terms.
- R_{GO} is the set of relationships, referring to relationships between GO terms.

Since there are two elements in GO, the change operations can be applied to both elements, i.e. concepts and relationships. A concept can be deleted, added, or altered in term of the name. The formal definition of addition of concept in GO as shown by Definition 2.

Definition 2: Addition of a GO concept

$$AddCon(c_{new}, O_{GO}) \Leftrightarrow O_{GO} | C_{GO} \leftarrow C_{GO} \cup \{c_{new}\}$$

In this case, c_{new} is the new concept that is added to the set of concepts C_{GO} in GO.

The definition of deletion of a concept in GO is shown by Definition 3.

Definition 3: Deletion of a GO concept

$$DelCon(c_{del}, O_{GO}) \Leftrightarrow O_{GO} | C_{GO} \leftarrow C_{GO} - \{c_{del}\}$$

c_{del} represents the concept to be deleted from the set of concepts C_{GO} in GO.

The name of a concept can be altered. The definition of alteration of a concept name in GO is shown by Definition 4.

Definition 4: Alteration of a GO concept name

$$AltCon(c_{GO}, newname) \Leftrightarrow c_{GO} | name(c_{GO}) \leftarrow newname$$

From the above definition, $name(c_{GO})$ refers to the name attribute of concept c in GO and is assigned with a new value, which is $newname$.

For relationship, there are two operations that can be applied, i.e. addition and deletion. Addition of relationship in GO is mostly caused by the addition of a new concept. The formal representation of addition of a relationship in GO is shown by Definition 5.

Definition 5: Addition of a GO relationship

$$AddRel(r_{new}, O_{GO}) \Leftrightarrow O_{GO} | R_{GO} \leftarrow R_{GO} \cup \{r_{new}\}$$

From the definition, r_{new} is the new relationship that is added to the set of relationship R_{GO} in GO.

The definition of deletion of a relationship in GO is shown by Definition 6.

Definition 6: Deletion of a GO relationship

$$DelRel(r_{del}, O_{GO}) \Leftrightarrow O_{GO} | R_{GO} \leftarrow R_{GO} - \{r_{del}\}$$

r_{del} represents the relationship to be deleted from the set of relationships R_{GO} in GO.

Based on the above formal representations of GO change operations, the classification of change operations in GO is shown by Fig. 2.

5.2 Change operations in MeSH

MeSH has more elements that must be considered than GO. The elements are the Descriptors, the relationships between Descriptors that form a tree of Descriptors, the Qualifiers, and the relationship between the Descriptors and the Qualifiers. The formal definition of MeSH ontology is shown by Definition 7.

Definition 7: MeSH Ontology

$O_M \equiv \langle C_M, R_M, Q_M, M_M \rangle$ is GO ontology with:

- C_M is the set of MeSH concepts, referring to MeSH Descriptors.

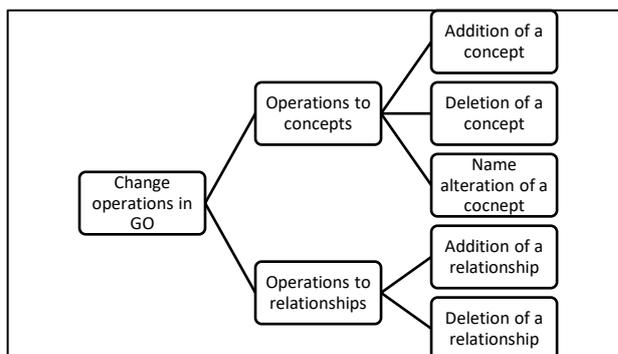


Figure. 2 Classifications of change operations in GO

- R_M is the set of relationships, referring to relationships between MeSH Descriptors that form a hierarchy consisting of Descriptors. The type of relationship is IS A.
- Q_M is the set of qualifiers, referring to MeSH Qualifiers.
- M_M is the set of mappings, referring to relationships between Descriptors and Qualifiers.

Change operations in MeSH can be applied to all of its elements, i.e. concepts, relationships, qualifiers and mappings. Similar to a GO concept, a MeSH concept can also be added, deleted, and changed in terms of its name. Definition 8, 9 and 10 represent the formal definition of addition of concept, deletion of concept, and name change of a concept, respectively.

Definition 8: Addition of a MeSH concept

$$AddCon(c_{new}, O_M) \Leftrightarrow O_M | C_M \leftarrow C_M \cup \{c_{new}\}$$

Definition 9: Deletion of a MeSH concept

$$DelCon(c_{del}, O_M) \Leftrightarrow O_M | C_M \leftarrow C_M - \{c_{del}\}$$

Definition 10: Alteration of a MeSH concept name

$$AltCon(c_M, newname) \Leftrightarrow c_M | name(c_M) \leftarrow newname$$

In this case, c_{new} is the new concept that is added to the set of MeSH concepts C_M , c_{del} represents the concept to be deleted from C_M , while $name(c_M)$ refers to the name attribute of concept c_M that is assigned with a new value, which is *newname*.

Change operations to a MeSH qualifier are basically the same as change operations for MeSH concepts. Hence, a qualifier can be added, deleted, and changed in terms of its name. Definition 11, 12 and 13 represent the formal definition of addition of qualifier, deletion of qualifier, and name change of a qualifier, respectively.

Definition 11: Addition of a MeSH qualifier

$$AddCon(q_{new}, O_M) \Leftrightarrow O_M | Q_M \leftarrow Q_M \cup \{q_{new}\}$$

Definition 12: Deletion of a MeSH qualifier

$$DelCon(q_{del}, O_M) \Leftrightarrow O_M | Q_M \leftarrow Q_M - \{q_{del}\}$$

Definition 13: Alteration of a MeSH qualifier name

$$AltQual(q_M, newname) \Leftrightarrow q_M | name(q_M) \leftarrow newname$$

In the above definitions, q_{new} is the new qualifier that is added to the set of qualifiers Q_M , q_{del} represents the qualifier to be deleted from the set of qualifiers Q_M in MeSH, while $name(q_M)$ refers to

the name attribute of qualifier q_M that is assigned with a new value, which is *newname*.

For both relationships and mappings, there are two operations that can be applied, i.e. addition and deletion. The formal representation of addition of a relationship, deletion of a relationship, addition of a mapping, and deletion of a mapping in MeSH are shown by Definition 14, 15, 16 and 17, respectively.

Definition 14: Addition of a MeSH relationship

$$AddRel(r_{new}, O_M) \Leftrightarrow O_M | R_M \leftarrow R_M \cup \{r_{new}\}$$

Definition 15: Deletion of a MeSH relationship

$$DelRel(r_{del}, O_M) \Leftrightarrow O_M | R_M \leftarrow R_M - \{r_{del}\}$$

Definition 16: Addition of a MeSH mapping

$$AddMap(m_{new}, O_M) \Leftrightarrow O_M | M_M \leftarrow M_M \cup \{m_{new}\}$$

Definition 17: Deletion of a MeSH mapping

$$DelMap(m_{del}, O_M) \Leftrightarrow O_M | M_M \leftarrow M_M - \{m_{del}\}$$

From the definitions, r_{new} is the new relationship to be added to the set of relationship R_M , r_{del} is the relationship to be deleted, m_{new} is the new mapping to be added to the set of mappings M_M , m_{del} is the mapping to be deleted from MeSH.

Based on the above formal representations of MeSH change operations, the classification of change operations in MeSH is shown by Fig. 3.

6. Mapping of change operations from GO into MeSH

The characteristics of change operations in GO and MeSH can be compared based on the formal representation and classification of change operations in both ontologies.

Since the number of elements in both ontologies are not the same, not all change operations can be used in the mapping process. There are only two elements in GO, i.e. concepts and relationships, while there are four elements in MeSH, i.e. concepts, qualifiers, relationships and mappings. To simplify the mappings, in this paper, we only consider the elements that appear in both ontologies: concepts and relationships.

There are three change operations that can be applied to GO concepts as well as MeSH concepts, i.e. addition, deletion and name alteration. For relationships, two types of operations are available in GO and MeSH, i.e. addition and deletion. Since the types of operation for each element in both ontologies are similar, it is expected that the mapping of change operations between both ontologies can be performed. Before performing the mapping of change operations,

Table 3. Comparison of change operation semantic meaning in GO and MeSH

Elements	Operations	Gene Ontology		MeSH	
		Condition before change	Condition after change	Condition after change	Condition after change
Concept	Addition	C_{GO}	$C_{GO} \leftarrow C_{GO} \cup \{c_{new}\}$	C_M	$C_M \leftarrow C_M \cup \{c_{new}\}$
	Deletion	C_{GO}	$C_{GO} \leftarrow C_{GO} - \{c_{del}\}$	C_M	$C_M \leftarrow C_M - \{c_{del}\}$
	Name alteration	$name(c_{GO}) \leftarrow oldname$	$name(c_{GO}) \leftarrow newname$	$name(c_M) \leftarrow oldname$	$name(c_M) \leftarrow newname$
Relationship	Addition	R_{GO}	$R_{GO} \leftarrow R_{GO} \cup \{r_{new}\}$	R_M	$R_M \leftarrow M \cup \{r_{new}\}$
	Deletion	R_{GO}	$R_{GO} \leftarrow R_{GO} - \{r_{del}\}$	R_M	$R_M \leftarrow R_M - \{r_{del}\}$

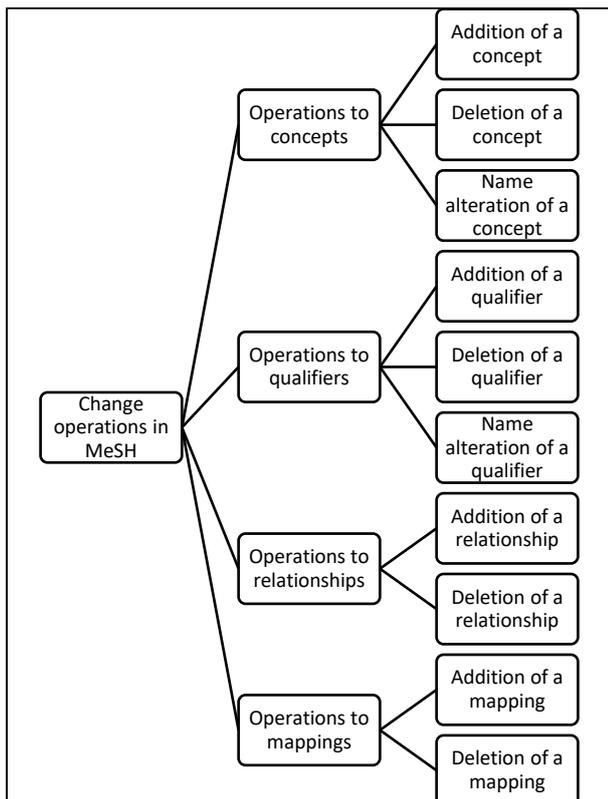


Figure. 3 Classifications of change operations in MeSH

Table 4. Mapping of change operations from GO into MeSH

GO change operation	MeSH change operation
$AddCon(c_{new}, O_{GO})$	$AddCon(c_{new}, O_M)$
$DelCon(c_{del}, O_{GO})$	$DelCon(c_{del}, O_M)$
$AltCon(c_{GO}, newname)$	$AltCon(c_M, newname)$
$AddRel(r_{new}, O_{GO})$	$AddRel(r_{new}, O_M)$
$DelRel(r_{del}, O_{GO})$	$DelRel(r_{del}, O_M)$

the semantic meaning of change operations should be examined. To compare the semantic meaning of change operations in GO and in MeSH, the required condition before and after change must be identified.

Table 3 presents the comparison of the semantic meaning of change operations in both ontologies. From the table, it can be seen that for each type of

change operation, the condition before and after the change in both ontologies are identical. Hence, it can be concluded that each change operation in GO can be mapped into a corresponding operation in MeSH.

Table 4 shows the mapping of change operations from GO into MeSH. In the table, a GO concept is mapped into a MeSH concept addition, a GO concept deletion is mapped into a MeSH concept deletion, and GO name alteration of concept is mapped into a MeSH name alteration of concept.

The mapping of relationship is conducted in a similar way. A GO addition of relationship is mapped into MeSH addition of relationship, while GO deletion of relationship is mapped into MeSH deletion of relationship.

Fig. 4 describes the mapping process from GO to MeSH. The mapping is specifically for deletion and name alteration on concepts. The input is the GO change operations. This means that the before mapping process is started, change operations in GO must be identified first. The change operations from the input is processed one by one. The changed concept is then found in UMLS. If there is a corresponding MeSH concept in UMLS, the change operation is mapped into MeSH by creating the corresponding change operation in MeSH. If not, next change operation is processed until all change operations have been processed. The output is the list of MeSH change operations.

7. Evaluation and discussion

In this chapter, we present an example of the application of a change operation mapping from GO to MeSH. Then, several issues related to the method will be discussed.

7.1 A proof of concept

The concepts included in GO are specifically related to gene functions, while the concepts in MeSH are more general. Changes between two recent consecutive releases of GO involve concepts that are

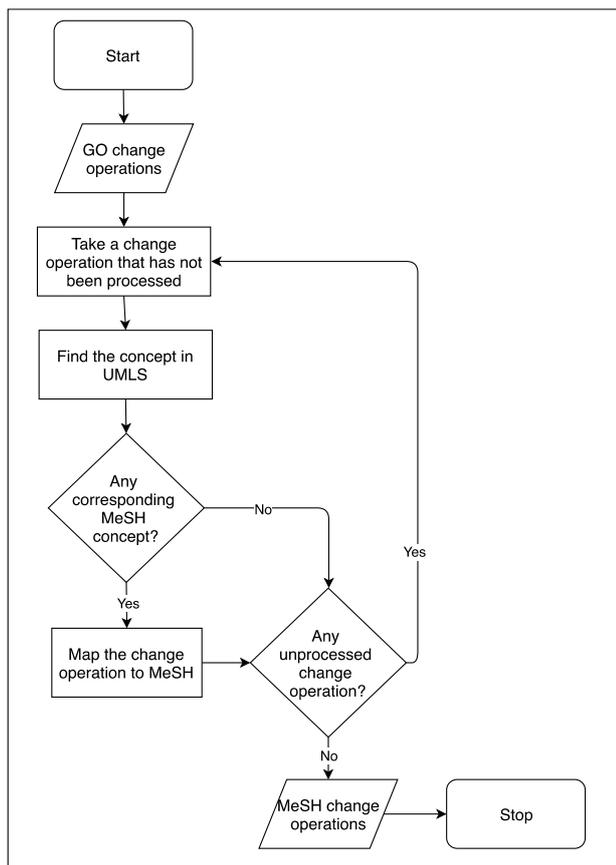


Figure. 4 Flowchart of the mapping process for deletion and name alteration on concepts

specifically included in GO but not in MeSH. For this reason, we could not use the changes between two consecutive releases of GO to show how our proposed method works. Instead, we examined the 2019-10 GO release to find the deleted (obsolete) concepts in the release when compared with the 2016-10 release. For each deleted concept, we inspected UMLS to check whether there is a corresponding MeSH concept. If it existed, the GO change operation is mapped into MeSH by creating the corresponding MeSH change operation.

One concept that has been changed in GO is *mutagenesis* with concept ID GO:0006280. From this change, a concept deletion change operation was created. The change operation is $DelCon(GO:0006280, O_{GO})$. The concept is available in UMLS with Concept Unique Identifier (CUI) of C0079866 with String Unique Identifier (SUI) of *Mutagenesis Process*. From UMLS, it can be seen that the concept also refers to a MeSH concept with the unique ID D016296. Apparently, the concept still exists in MeSH with the name *Mutagenesis*. This means that the change that happened in GO has not been propagated into MeSH. Thus, a new MeSH change operation must be created.

Table 5. Mapping of change operations from GO into MeSH

Deleted GO concept	Relevant UMLS concept(s)	Affected MeSH concept
GO:0006280	C0079866	D016296
GO:0000988	C0033618	D011485
GO:0002146	C3154910, C0034840	D011987
GO:0002147	C3154911, C0034809	D011965
GO:0003840	C1324645, C0017040	D005723
GO:0003840	C3896387, C0017040	D005723
GO:1990602	C3896387, C0288912	D028901
GO:1902376	C3822175, C0033627	D011489
GO:1902002	C3546774	-
GO:1902592	C3822809	-

The corresponding MeSH change operation is $DelCon(D016296, O_M)$.

The example shows that our proposed method works well to map a change operation in GO into a change operation in MeSH. This can be used to identify which MeSH concept is infected due to a change that has happened in GO.

7.2 Experiment results

The proposed method was applied to ten different change operations in GO, among hundreds of deletion operations that can be found from the 2016-10 GO release to the 2016-10 release. Since deletion of concepts is the operation that is easiest to be detected, we used deletion operations to test how our method worked. The ten operations were chosen because they have appropriately represented the three possible situations that can be found when applying our proposed method. Table 5 shows the result of the experiment. For each deleted concept, we found the relevant UMLS concept(s). Using the relevant UMLS concept(s), we tried to find the affected MeSH concept due to the deletion of GO concept. The affected MeSH concept is the concept that needs to be considered to be deleted to make the MeSH updated with the change that happens in GO.

As previously mentioned, there are three possibilities that can be faced when a GO concept deletion operation is propagated to MeSH. In each of the first two rows of Table 3, exactly one relevant UMLS concept is found, that represents both the deleted GO concept and the corresponding MeSH concept. This is the ideal situation, in which the GO deletion operation can be directly propagated to a

MeSH deletion operation. For the next six rows, there is no unique UMLS concept to represent both GO concept and MeSH concept. In this case, we have to find similar concepts in UMLS that represent relevant MeSH concepts. A careful decision must be made to determine whether the GO deletion operation must be propagated into a MeSH deletion operation or not, because it is possible that the relevant MeSH concept does not represent the same semantic as the deleted GO concept. In the last two rows, there is no similar UMLS concept that represents a corresponding MeSH concept, hence, the operation cannot be propagated into a MeSH operation. The lack of relevant MeSH concept is usually due to the specificity of the GO concept. For example, GO ID concept of GO:1902002 represents multi-organism membrane budding concept, which is not available in MeSH.

7.3 Discussion

As stated previously, the proposed method of the mapping of change operations can be applied well to map a GO change operation into a MeSH change operation. A change operation in GO, especially a deletion or name alteration of a concept, indicates that the corresponding MeSH concept, if available, should be changed as well. However, since GO concept tends to be more specific related to gene terms, it is possible that a GO concept deletion operation cannot be propagated into a MeSH concept deletion operation because there is no MeSH concept that represents the GO deleted concept. When the UMLS concept that correspond to the GO deleted concept does not represent a MeSH concept, another UMLS concept that represents a relevant MeSH concept must be sought. In this case, it must be examined first whether the relevant MeSH concept should be deleted or not.

A GO concept deletion or name alteration can be mapped directly into a MeSH concept deletion or name alteration because they are applied to existing concepts. This is the reason why we use a deletion operation to show how our method works. The same approach cannot be applied to concept addition. Since GO is more specific than MeSH, it is expected that not every GO concept addition can be mapped into an MeSH concept addition. In this case, an examination should be performed by MeSH expert to decide whether the new GO concept should be also be added into MeSH.

For change operations on relationship, the case is tricky. The only type of relationship to which the method can be applied is the `IS A` relationship, because MeSH hierarchy only involves this

relationship type. However, to map a GO deletion of relationship into MeSH we have to check first whether the GO relationship is also available in MeSH with the corresponding MeSH concepts as the GO concepts that are involved in the GO relationship. Even though this is possible, we found out that such exact relationship is very hard to find. This also applies to relationship addition, in which it must be examined whether there are corresponding MeSH concepts that are available for the GO concepts that are involved in the relationship.

It is possible that the relationship between a MeSH Description and a MeSH Qualifier can be considered as a type of relationship. In this case, a Qualifier should be regarded as an ontology concept. However, since the corresponding relationship type in GO does not exist explicitly, we do not use this approach in this paper.

As previously stated, GO concepts are much more specific than MeSH concepts. It is understandable that the intersection between the GO concepts and the MeSH concepts is small. This is the reason that we could not find many UMLS concepts which source from both GO and MeSH. This is one of the issues that must be considered when we do the mapping of change operations from one ontology into another. Ideally, the intersection between ontologies involved in the mapping should be large so that the mapping process gives the optimum results.

Even though our method is applied only to GO and MeSH, it can also be applied to other pair of ontologies. For example, we can do the same procedure from SNOMED CT to MeSH or other ontology. We only need to identify the change operations in each ontology, find the characteristics of each change operation and do the mapping. With this mapping, it is expected that the target ontology will be kept up-to-date because it always kept informed with the changes in the source ontology.

8. Conclusion

A method for change propagation has been proposed in this paper to address the issue of difficulty in maintaining the mapping between two different ontologies due to the changes happen in one of the ontologies. The change propagation is applied to a pair of ontologies, i.e. GO and MeSH, that have common elements. Changes are propagated by performing a mapping of change operation from the source ontology to the target ontology. A proof of concept has been presented to show how the approach works. In the experiment, the method is able to propagate 80% of the GO concept deletion operations to MeSH. Some operations cannot be propagated

because for each of them, there is no corresponding MeSH concept due to the specificity of the concept. With this propagation, the recency of the target ontology, i.e. MeSH can be maintained.

For future work, the mapping of change operations can be conducted to different pair or ontologies. The results will be optimum if the set of concepts intersection between both ontologies is large.

References

- [1] Regenstrief Institute, “About LOINC”, 2020. [Online]. Available: <https://loinc.org/about/>. [Accessed: 21-Jan-2020].
- [2] National Library of Medicine, “Medical Subject Headings”, 2019. [Online]. Available: <https://www.nlm.nih.gov/mesh/meshhome.html>. [Accessed: 21-Jan-2020].
- [3] G. O. Consortium, “About the GO”, 2020. [Online]. Available: <http://geneontology.org/docs/introduction-to-go-resource/>. [Accessed: 21-Jan-2020].
- [4] OpenEHR, “Archetype Definition Language 2 (ADL2)”, 2019. [Online]. Available: <https://specifications.openehr.org/releases/AM/atest/ADL2.html>. [Accessed: 10-Aug-2019].
- [5] F. Minguet, T. M. Salgado, L. van den Boogerd, and F. Fernandez-Llimos, “Quality of pharmacy-specific Medical Subject Headings (MeSH) assignment in pharmacy journals indexed in MEDLINE”, *Res. Soc. Adm. Pharm.*, Vol. 11, No. 5, 2015, pp. 686–695, 2015.
- [6] Y. Zhao, G. Fu, J. Wang, M. Guo, and G. Yu, “Gene function prediction based on Gene Ontology Hierarchy Preserving Hashing”, *Genomics*, Vol. 111, No. 3, pp. 334–342, 2019.
- [7] G. Agapito, M. Milano, P.H. Guzzi and M. Cannataro, “Extracting Cross-Ontology Weighted Association Rules from Gene Ontology Annotations”, *IEEE/ACM Trans. Comput. Biol. Bioinform.*, Vol 13, No. 2, pp. 197-208, 2016.
- [8] K. W. Fung and J. Xu, “Synergism between the Mapping Projects from SNOMED CT to ICD-10 and ICD-10-CM”, In: *American Medical Informatics Association Annual Symposium Proceeding*, pp. 218–227, 2012.
- [9] A. Maedche, B. Motik, and L. Stojanovic, “Managing multiple and distributed ontologies on the semantic web”, *VLDB J.*, Vol. 12, pp. 286–302, 2003.
- [10] R. Palma, O. Corcho, A. Gmez-Prez, and P. Haase, “A holistic approach to collaborative ontology development based on change management”, *J. Web Semant.*, Vol. 9, No. 3, pp. 299–314, 2011.
- [11] R. Zhang, D. Guo, W. Gao, and L. Liu, “Modeling ontology evolution via Pi-Calculus”, *Inf. Sci. (Ny)*, Vol. 346–347, pp. 286–301, 2016.
- [12] A. K. Sari, “Identification and formal representation of change operations in LOINC evolution”, *Int. J. Adv. Comput. Sci. Appl.*, Vol. 10, No. 1, 2019.
- [13] H. Kondylakis, M. Despoina, G. Glykokokalos, E. Kalykakis, M. Karapiperakis, M.-A. Lasithiotakis, J. Makridis, P. Moraitis, A. Panteri, M. Plevraki, A. Providakis, M. Skalidaki, A. Stefanos, M. Tampouratzis, E. Trivizakis, F. Zervakis, E. Zervouraki, and N. Papadakis “EvoRDF: A Framework for Exploring Ontology Evolution”, In: *Proc. of European Semantic Web Conference ESWC 2017: The Semantic Web*, pp. 104–108, 2017.
- [14] Z. Liu and L. Yin, “Research on the Evolution Method of Domain Ontology Based on Dbpedia”, *Wirel. Pers. Commun.*, Vol. 102, No. 2, pp. 1735–1749, 2018.
- [15] S. D. Cardoso, M. Da Silveira, and C. Pruski, “Construction and exploitation of an historical knowledge graph to deal with the evolution of ontologies”, *Knowledge-Based Syst.*, 2020. DOI: 10.1016/j.knosys.2020.105508.
- [16] C. Huang, H. Cai, L. Xu, B. Xu, Y. Gu, and L. Jiang, “Data-driven ontology generation and evolution towards intelligent service in manufacturing systems”, *Futur. Gener. Comput. Syst.*, Vol. 101, pp. 197–207, 2019.
- [17] M. Gawich, M. Alfonse, M. Aref, and A.-B. M. Salem, “Ontology Maintenance System for Rheumatoid Disease”, *Procedia Comput. Sci.*, Vol. 154, pp. 241–346, 2018.
- [18] J. Dos Reis, “Mapping refinement based on ontology evolution: A context-matching approach”, In: *CEUR Workshop Proceedings*, Vol. 2228, pp. 251–256, 2018.
- [19] J. M. Destro, J. C. dos Reis, R. da Silva Torres, and I. Ricarte, “Ontology changes-driven semantic refinement of cross-language biomedical ontology alignments”, In: *Proc. of Workshop on Semantic Web Solutions for Large-Scale Biomedical Data Analytics 2019*, pp. 31–15, 2019.
- [20] F. Antoniazzi and F. Viola, “Building the Semantic Web of Things Through a Dynamic Ontology”, *IEEE Internet Things J.*, Vol. 6, No. 6, pp. 10560–10579, 2019.
- [21] Y.-H. Liu and N. Wacholder, “Evaluating the impact of MeSH (Medical Subject Headings)

- terms on different types of searchers”, *Inf. Process. Manag.*, Vol. 53, pp. 851–870, 2017.
- [22] A. M. Petersen, D. Rotolo, and L. Leydesdorff, “A triple helix model of medical innovation: Supply, demand, and technological capabilities in terms of Medical Subject Headings”, *Res. Policy*, Vol. 45, pp. 666–681, 2016.
- [23] D. Hongyong, C. E. Adams, F. Shokrane, and L. Shanghua, “Classification of interventions in Traditional Chinese Medicine”, *J. Tradit. Chinese Med.*, Vol. 38, pp. 315–320, 2018.
- [24] G. Tsatsaronis, I. Varlamis, N. Kanhabua, and K. Nørvi, “Temporal Classifiers for Predicting the Expansion”, In: *Proc. of the 14th Int. Conf. Intell. Text Process. Comput. Linguist.*, pp. 98–113, 2013.
- [25] K. Vanopstal, J. Buysschaert, G. Laureys, and R. Vander Stichele, “Lost in PubMed. Factors influencing the success of medical information retrieval”, *Expert Syst. Appl.*, Vol. 40, No. 10, pp. 4106–4114, 2013.
- [26] G. Yu, Y. Zhao, C. Lu, and J. Wang, “HashGO: hashing gene ontology for protein function prediction”, *Comput. Biol. Chem.*, Vol. 71, pp. 264–273, 2017.
- [27] S. Acharya, S. Saha, and P. Pradhan, “Novel symmetry-based gene-gene dissimilarity measures utilizing Gene Ontology: Application in gene clustering”, *Gene*, Vol. 67930, pp. 341–351, 2018.
- [28] A. K. Paul and P. C. Shill, “Incorporating gene ontology into fuzzy relational clustering of microarray gene expression data”, *Biosystems*, Vol. 163, pp. 1–10, 2018.