CoVoMe: New Methodology for Building Controlled Vocabulary

Dominik Tomaszuk(✉)

Institute of Computer Science, University of Bialystok, Bialystok, Poland
d.tomaszuk@uwb.edu.pl

Abstract. The use of methodologies in knowledge management and engineering is deeply comprehensive due to their important advantages. In this paper, we propose CoVoMe that is a methodology for building controlled vocabularies. This methodology covers almost all variants of that vocabularies, and it is designed to be close to the currently available languages for creating thesauri, subject headings, taxonomies, authority files, synonym rings, and glossaries.

Keywords: Knowledge organization system · Controlled vocabulary · Methodology

1 Introduction

The term knowledge organization system (KOS) is intended to cover all types of controlled vocabularies (CVs) for organizing information and promoting knowledge management. Compared to free-text searching, the use of a CV can greatly increase the performance and precision of a system. Controlled vocabularies are used in different domains, e.g., libraries [8,27], medicine [26], food [7], art [37], economy [30], etc.

A lot of CVs have been developed by different groups of people, under different approaches, and using different methods and techniques. Unfortunately, there are not too many well-documented activities, life cycles, standardized methodologies, and well-defined design criteria. On the other hand, there are many methodologies for ontologies [14,33,39,40]. At the same time, there are only a few similar, but not so complex, proposals to thesauri, taxonomies, or other controlled vocabularies that support the above features (see the related work in Sect. 4). Moreover, there are no proposals that cover all variants of CVs. The field of CV construction still lacks standardized methodologies that can be adapted to different conditions. The major cause is that most of the methodologies were applied to develop CVs for specific projects and/or types of CV. So, the generalization of the methodology was not proposed for other contexts. In this paper, we propose CoVoMe which is a methodology for building CVs either from scratch or reusing by a process of re-engineering them. This methodology covers almost all variants of the controlled vocabularies. Moreover, it is designed to be close to the currently available languages for creating CVs.

E. Garoufallou et al. (Eds.): MTSR 2021, CCIS 1537, pp. 41–56, 2022.
https://doi.org/10.1007/978-3-030-98876-0_4

The paper is organized as follows. Section 2 contains basic definitions used throughout this paper. In Sect. 3, we describe our methodology. In Sect. 4, we discuss related work. Finally, in Sect. 5, we summarize our findings and outline further research directions.

2 Preliminaries

Controlled vocabularies are used in different forms, such as thesauri [7,26,30], classification schemes [27,28], subject headings [8], taxonomies [10], authority files [16], etc.

A controlled vocabulary is a standardized and organized arrangement of words and phrases used to retrieve content through searching and provide a consistent way to describe data. Metadata and data providers assign terms from vocabularies to improve information retrieval. It should typically have a defined scope or describe a specific domain. In this paper, we define full controlled vocabulary as a broad term. The full controlled vocabulary abstraction is defined to be compatible with all kinds of CVs.

Definition 1 (Full controlled vocabularies). *A* full controlled vocabulary *is defined as a tuple of the form* $V = \langle RS, C, CS, SR, MP, LD, CO \rangle$*, where*

1. *RS is the set of resources,*
2. *$C \subseteq RS$ is the set of concepts, which are all concepts that are identified by IRIs in the vocabulary namespace,*
3. *CS is the set of concept schemes that aggregate concepts,*
4. *SR is the set of semantic relations that include relations for hierarchies (RH) and relation for association (RA),*
5. *MP is the set of mapping properties that includes properties for hierarchy mapping (HM), association mapping (AM) and similarity (PS) and associate resources with one another,*
6. *LD is the set of labels (L), notation (N), and documentation properties (D),*
7. *CO is the set of unordered and ordered collections.*

Full controlled vocabularies are the basis for other definitions. We start with a simple glossary and end with an advanced thesaurus.

We define a glossary as an alphabetical list of terms, usually in a specific domain with the definitions for those terms.

Definition 2 (Glossaries). *A* glossary *is defined as a tuple of the form* $G = \langle RS, C, LD \rangle$.

A slightly more expanded form is a synonym ring (also called synset). We define it as a group of terms that are considered semantically equivalent for the purpose of retrieval.

Definition 3 (Synonym rings). *A* synonym ring *is defined as a tuple of the form* $R = \langle RS, C, RA, LD \rangle$.

Then, we define authority files. An authority file is lists of terms that are used to control the variant names for an object for a particular area. They are also applied to other methods of organizing data such as linkages and cross-references.

Definition 4 (Authority files). *An* authority file *is defined as a tuple of the form* $A = \langle RS, C, CS, PS, LD \rangle$.

We define a taxonomy as the division of items into categories or classifications, especially a hierarchical classification, based on particular characteristics.

Definition 5 (Taxonomies). *A* taxonomy *is defined as a tuple of the form* $T = \langle RS, C, CS, RH, LD \rangle$.

Subject heading is slightly more complicated. It provides a group of terms to represent the subjects of items in a collection and sets of rules for connecting terms into headings.

Definition 6 (Subject headings). *A* subject heading *is defined as a tuple of the form* $H = \langle RS, C, CS, SR, LD \rangle$.

The quite complex form of controlled vocabularies is a thesaurus. We define a thesaurus as collections of terms representing concepts and the hierarchical, equivalence, and associative relationships among them.

Definition 7 (Thesauri). *A* thesaurus *is defined as a tuple of the form* $S = \langle RS, C, CS, SR, LD, CO \rangle$.

Table 1 presents a summary of the characteristics of the above-defined controlled vocabularies.

Table 1. Features of controlled vocabularies

		Glossaries	Synonym rings	Authority files	Taxonomies	Subject headings	Thesauri	Full controlled vocabulary
Concepts and schemas		y	y	y	y	y	y	y
Lab. and notation		y	y	y	y	y	y	y
Documentation		y	y	y	y	y	y	y
Sem. relations	hrchy.	n	n	n	n	y	y	y
	assoc.	n	y	n	n	y	y	y
Map. properties	hrchy.	n	n	n	y	n	n	y
	assoc.	n	n	n	n	n	n	y
	sim.	n	n	y	n	n	n	y
Collections		n	n	n	n	n	y	y

3 Methodology and Steps

A CoVoMe methodology has eight steps and some of the steps are divided into activities. CoVoMe consists of the following steps:

Step 1: Determine the domain and scope (Subsect. 3.1),
Step 2: Determine the type of controlled vocabulary (Subsect. 3.2),
Step 3: Define the concepts and concept schemas (Subsect. 3.3),
Step 4: Define the terms, labels and notation (Subsect. 3.4),
Step 5: Define the semantic relations (Subsect. 3.5),
Step 6: Define groups of concepts (Subsect. 3.6),
Step 7: Integrate with other controlled vocabularies (Subsect. 3.7),
Step 8: Create the documentation (Subsect. 3.8).

In Subsect. 3.9, we discuss how to evaluate our proposal, and how the chackpoints are connected to CoVoMe steps. Figure 1 shows the steps and activities order.

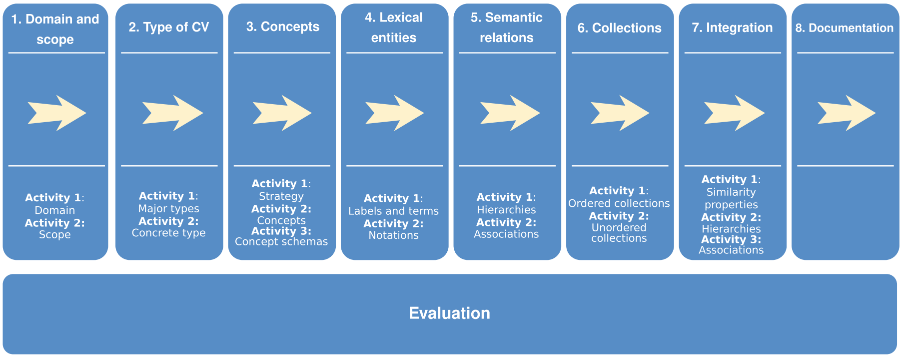

Fig. 1. Sequence of steps in CoVoMe

3.1 Determine the Domain and Scope

In order to define scope, the user should go through two activities. The first one specify some sources that could be used to acquire knowledge for the CV development. In the second activity, a user should use competency questions (CQs) [17] to determine the scope.

In the first activity, a user should acquire a domain knowledge from several sources, such as domain experts, domain literature, other controlled vocabularies, etc. One can also use different techniques, e.g. interviews, brainstorming, mind maps, etc.

In the second activity, based on the first one, the recommended way to determine the scope of the CoVoMe is to sketch a list of CQs that one should be able to answer. CQs are natural language questions outlining and constraining the scope of knowledge represented in a vocabulary. Note that the answers to these questions may change during the process, but at any given time they help limit the scope of the model.

3.2 Determine the Type of Controlled Vocabulary

In this step, one needs to determine what type of controlled vocabulary will be constructed. At this point, users specifies what type of KOS they are building. The first activity at this step is to choose major types of a CV. These major types are based on features such as structure, complexity, relationships among terms, and historical function. According to [19], one can choose three following options:

1. term lists: a CV that emphasizes lists of terms often with definitions,
2. classifications and categories: a CV that emphasizes the creation of subject sets,
3. relationship lists: a CV that emphasizes the connections between terms and concepts.

In the second activity, users should choose the concrete type of CV. According to the definitions in Sect. 2, one can choose:

1. in term lists: glossary, synonym ring or authority file,
2. in classifications and categories: taxonomy or subject headings,
3. in relationship lists: thesaurus or full controlled vocabulary.

Classifications established previously [38] may also be helpful for users in this activity. Note that the selection of a specific type will affect the next steps, e.g. if the user has selected taxonomy, he must complete step 3, 4 and 8, partially step 5, but steps 6 and 7 do not apply to her/him.

In the third activity of this stage in CoVoMe, one should choose vocabulary for building the CV. In this paper, in the steps 3–8, we use SKOS [5], ISO 25964 [21], and MADS [29] that are the most popular. However users are not limited to these vocabularies. Additionally, we suggest which vocabulary elements for building CV may be used in a given step.

3.3 Define the Concepts and Concept Schemas

Partial support: Glossaries, Synonym rings,
Full support: Authority files, Taxonomies, Subject headings, Thesauri, Full controlled vocabularies.

SKOS vocabulary: `dcterms:hasPart`[1], `skos:Concept`, `skos:ConceptScheme`, `skos:hasTopConcept`, `skos:inScheme`, `skos:topConceptOf`,
ISO 25964 vocabulary: `iso25964:CustomConceptAttribute`, `iso25964:CustomTermAttribute`, `iso25964:Thesaurus`, `iso25964:ThesaurusConcept`, `iso25964:TopLevelRelationship`, `iso25964:contains`, `iso25964:isPartOf`,

[1] `dcterms:hasPart` is not a part of SKOS but sometimes is used to define coordinations.

MADS vocabulary: `mads:Authority`, `mads:ComplexType`, `mads:DeprecatedAuthority`, `mads:MADSScheme`, `mads:MADSType`, `mads:RWO`, `mads:SimpleType`, `mads:hasMADSSchemeMember`, `mads:hasTopMemberOfMADSScheme`, `mads:identifiesRWO`, `mads:isTopMemberOfMADSScheme`.

In this step, one should denominate ideas, meanings, or objects and events. In the first activity, users should choose strategy for identifying the concepts. According to [15], users can decide which option to choose:

– bottom-up: start from the most specific concepts and build a structure by generalization,
– top-down: start from the most generic concept and build a structure by specialization,
– middle-out: core concepts are identified and then generalised and specialised to a complete list.

In the second activity, one should define the concepts according to the previously selected strategy. In the last step, users should organize and aggregate concepts into concept schemes.

3.4 Define the Terms, Labels and Notation

Full support: Glossaries, Synonym rings, Authority files, Taxonomies, Subject headings, Thesauri, Full controlled vocabularies.

SKOS vocabulary: `skos-xl:Label`[2], `skos-xl:altLabel` (see footnote 2), `skos-xl:hidden- Label` (see footnote 2), `skos-xl:prefLabel` (see footnote 2), `skos:altLabel`, `skos:hiddenLabel`, `skos:- notation`, `skos: prefLabel`,
ISO 25964 vocabulary: `iso25964:NodeLabel`, `iso25964:SimpleNonPreferredTerm`, `iso25964:SplitNonPreferredTerm`, `iso25964:ThesaurusTerm`, `iso25964:hasNodeLabel`, `iso25964:hasNonPreferredLabel`, `iso25964:hasPreferredLabel`, `iso25964:lexicalValue`, `iso25964:notation`,
MADS vocabulary: `mads:CorporateName`, `mads:Element`, `mads:Variant`, `mads:authoritativeLabel`, `mads:elementList`, `mads:elementValue`, `mads-:hasHiddenVariant`, `mads:hasVariant`.

This step allows, one to describe concepts, terms, and concept schemas in a way that people and machines can readily understand. The step allows for the description and link of lexical entities. This step can be divided into two activities. In the first activity, users should define human-readable labels/terms. Here it is possible to use different languages. In this activity the preferred string, maximum one per language tag, should be defined. Optionally users can define alternative strings.

In the next optional activity, one can define notations. Notations are helpful for classification codes and can be used to identify a concept within the scope of

[2] SKOS-XL is an extension for SKOS.

a given concept scheme, e.g., DD91.0Z can represent Irritable Bowel Syndrome in International Classification of Diseases revision 11. This activity is mainly dedicated machine-readable lexical codes.

3.5 Define the Semantic Relations

Partial support: Synonym rings, Taxonomies,
Full support: Subject headings, Thesauri, Full controlled vocabularies.

SKOS vocabulary: `dcterms:references`[3], `skos:broader`, `skos:broaderTransitive`, `skos:narrower`, `skos:narrowerTransitive`, `skos:related`, `skos:semanticRelation`,
ISO 25964 vocabulary: `gvp:broaderGeneric`[4], `gvp:broaderInstantial` (see footnote 4), `gvp:broaderPartitive` (see footnote 4), `iso25964:AssociativeRelationship`, `iso2596-4:CompoundEquivalence`, `iso25964:Equivalence`, `iso25964:Hierarchic-alRelationship`, `iso25964:broaderGeneric`, `iso25964:broaderInstant-ial`, `iso25964:broaderPartitive`, `iso25964:narrowerGeneric`, `iso2596-4:narrowerInstantial`, `iso25964:narrowerPartitive`, `iso25964:plusUF`, `iso25964:plusUSE`,
MADS vocabulary: `mads:hasBroaderAuthority`, `mads:hasEarlierEstablishedForm`, `mads:hasLaterEstablishedForm`, `mads:hasNarrowerAuthority`, `mads:hasReciprocalAuthority`, `mads:hasRelatedAuthority`, `mads:see`, `mads:useFor`, `mads:useInstead`.

This step defines ways to declare relationships between concepts within concept schemes. The step is divided into two activities. In the first activity, one should define relations for hierarchies, e.g. *narrower*, *broader* and its variants. Note that depending on the vocabulary used to build the CVs, there may be different deductive rules. Let $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$ be concepts, NT be a narrower relation, and BT be a broader relation, some of the following deductive rules that may be taken into account for this activity.

$$\frac{(\mathcal{C}_1\ NT\ \mathcal{C}_2)}{(\mathcal{C}_2\ BT\ \mathcal{C}_1)} \tag{1}$$

$$\frac{(\mathcal{C}_1\ BT\ \mathcal{C}_2)}{(\mathcal{C}_2\ NT\ \mathcal{C}_1)} \tag{2}$$

In some vocabularies, NT and BT can be transitive. Then the following rules are also possible.

$$\frac{(\mathcal{C}_1\ NT\ \mathcal{C}_2)\ (\mathcal{C}_2\ NT\ \mathcal{C}_3)}{(\mathcal{C}_1\ NT\ \mathcal{C}_3)} \tag{3}$$

$$\frac{(\mathcal{C}_1\ BT\ \mathcal{C}_2\ )\ (\mathcal{C}_2\ BT\ \mathcal{C}_3)}{(\mathcal{C}_1\ BT\ \mathcal{C}_3)} \tag{4}$$

[3] `dcterms:references` is not a part of SKOS but sometimes is used to define non-symmetric associative relations.
[4] GVP is an extension of ISO 25964 [1].

In the second activity, users should focus on relations for association, e.g. *related* and its variants. In the activity, the following deductive rule may be taken into account (*RT* is a related relation).

$$\frac{(\mathcal{C}_1\ RT\ \mathcal{C}_2)}{(\mathcal{C}_2\ RT\ \mathcal{C}_1)} \tag{5}$$

3.6 Define Groups of Concepts

Full support: Thesauri, Full controlled vocabularies.

SKOS vocabulary: `skos:Collection`, `skos:OrderedCollection`, `skos:member`, `skos:memberList`,
ISO 25964 vocabulary: `iso25964:ConceptGroup`, `iso25964:ConceptGroupLabel`, `iso25964:ThesaurusArray`, `iso25964:hasAsMember`, `iso25964:hasMemberArray`, `iso25964:hasMemberConcept`, `iso25964:hasSubgroup`, `iso25964:hasSubordinateArray`, `iso25964:hasSuperOrdinateConcept`, `iso25964:hasSupergroup`,
MADS vocabulary: `mads:Collection`, `mads:hasCollectionMember`, `mads:isMemberOfCollection`.

In this step, user defines groups of concepts that are useful where a collection of concepts have something in common, and it is convenient to group them. The collections can be nested. Depending on the vocabulary chosen for creating CVs, concept schemes can be usually part of a group, but semantic relations cannot apply to these groups.

This step is divided into two activities. In the first activity, a user may collect concepts and concept schemas that are ordered. In the next activity, one should check if the remaining entities can be grouped into unordered collections.

3.7 Integrate with Other Controlled Vocabularies

Partial support: Authority files,
Full support: Full controlled vocabularies.

SKOS vocabulary: `skos:broadMatch`, `skos:closeMatch`, `skos:exactMatch`, `skos:mappingRelation`, `skos:narrowMatch`, `skos:relatedMatch`,
MADS vocabulary: `mads:hasBroaderExternalAuthority`, `mads:hasCloseExternalAuthority`, `mads:hasCorporateParentAuthority`, `mads:hasCorporateSubsidiaryAuthority`, `mads:hasExactExternalAuthority`, `mads:hasNarrowerExternalAuthority`, `mads:hasReciprocalExternalAuthority`.

Some of the practices are acceptable according to the CVs, but having so many acceptable practices makes it more difficult for the consumer of an entity to find their way around. With the goal of standardization and indication of the similar objects in the construction, one might consider the reuse of resources already built into other CVs.

In this step, there are three activities. In the first one, a user defines similarity properties (exact or fuzzy mapping). Let $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ be concepts, EM be an exact relation, some of the following deductive rules that may be taken into account for this activity.

$$\frac{(\mathcal{C}_1\ EM\ \mathcal{C}_2)}{(\mathcal{C}_2\ EM\ \mathcal{C}_1)} \tag{6}$$

$$\frac{(\mathcal{C}_1\ EM\ \mathcal{C}_2\ )\ (\mathcal{C}_2\ EM\ \mathcal{C}_3)}{(\mathcal{C}_1\ EM\ \mathcal{C}_3)} \tag{7}$$

In the second activity, one can define hierarchy mapping properties, and in the last activity, mapping properties for association can be defined. Here, deductive rules that may be taken into account are analogous to a rule 1, a rule 2 (second activity), and a rule 5 (third activity). Note that this properties connect concepts from different schemes (in different CVs).

3.8 Create the Documentation

Full support: Glossaries, Synonym rings, Authority files, Taxonomies, Subject headings, Thesauri, Full controlled vocabularies.

SKOS vocabulary: `skos:definition`, `skos:editorialNote`, `skos:example`, `skos:historyNote`, `skos:note`, `skos:scopeNote`,
ISO 25964 vocabulary: `iso25964:CustomNote`, `iso25964:Definition`, `iso25964:EditorialNote`, `iso25964:HistoryNote`, `iso25964:Note`, `iso25964:ScopeNote`, `iso25964:VersionHistory`, `iso25964:hasCustomNote`, `iso25964:hasDefinition`, `iso25964:hasEditorialNote`, `iso25964:hasHistoryNote`, `iso25964:hasScopeNote`, `iso25964:refersTo`,
MADS vocabulary: `mads:changeNote`, `mads:definitionNote`, `mads:deletionNote`, `mads:editorialNote`, `mads:exampleNote`, `mads:historyNote`, `mads:note`, `mads:scopeNote`.

The goal of the documentation step is to catalog the development process and the CV itself. This step, including the maintenance, as well as definitions and examples should be embedded in the code of implemented CV. The languages for creating CVs often support different kinds of human-readable notes, e.g. explanation and information about the intended meaning of a concept, examples, information about historical changes, comments, etc.

3.9 Evaluation

At CoVoMe, we define an evaluation as a technical judgment of the CV and their environment during each step and activity. We distinguish between six different types of errors that can be found in each step:

- coverage level of the topic domain,
- check the completeness of the concepts,
- semantic inconsistency errors,
- lexical errors,
- circularity errors,
- redundancy detection.

Coverage Level of the Topic Domain. The extent to which a CV covers a considered domain is a crucial factor to be considered during the development process. The evaluation that can be employed to achieve this goal can be realized with similarity metrics [2]. This checkpoint is mostly dedicated to step 1 and step 2.

Check the Completeness of the Concept. The aim is to ascertain whether the concepts and/or concept schemas contain as much information as required. For example, errors appear when there are relations missing in the concept. This checkpoint is mostly dedicated to step 3 and step 5.

Semantic Inconsistency Errors. They usually occur because the user makes an incorrect semantic classification, that is, one classifies a concept as a semantic relation of a concept to which it does not really belong. For example, one classifies the *ornithology* concept as related to the *mammal* concept. This checkpoint is mostly dedicated to step 3 and step 6.

Lexical Errors. They occur when a label, a notation, a documentation property is not consistent with the data model because of the wrong value. For example, if we say that *animal* is a preferred label and at the same time *animal* is an alternative label, then the CV has a clash between the preferred and alternative lexical labels. An example rule to check if the preferred label (PL) is the same as the alternative label (AL) is presented below. This checkpoint is mostly dedicated to step 4 and step 8.

$$\frac{(x\ PL\ y)(x\ AL\ y)}{false} \tag{8}$$

Circularity Errors. They occur when a concept and/or concept scheme is defined as a specialization or generalization of itself. For example, if we say that *animal* is a narrower concept of *mammal*, and that *mammal* is a narrower concept of *animal*, then the CV has a circularity error. An example rule to check this error is presented below. This checkpoint is mostly dedicated to step 5 and step 7.

$$\frac{(x\ NT\ y)(y\ NT\ x)}{false} \tag{9}$$

Redundancy Detection. It occurs in CVs when there is more than one explicit definition of any of the hierarchical relations, or when we have two concepts with the same formal definition. For example, when a *dog* concept is defined as a broader concept of *mammal* and *animal*, and *mammal* is defined as a broader concept of *animal*, then, there is an indirect repetition. This checkpoint is mostly dedicated to step 5 and step 7.

4 Related Work

4.1 Construction of Controlled Vocabularies

Guidelines for the construction of controlled vocabularies have evolved over a long period. One of the first recommendations for building thesauri appeared in 1967 [18]. In this publication, the authors first defined terms such as *narrower*, *broader* and *related*. In the 1980s and 1990s, national [4,6,12] and international standards [20] for thesauri and controlled vocabularies were established. Other older guidelines for thesaurus construction have been reviewed by Krooks and Lancaster [25].

In [31], Nielsen analyzes the word association test and discusses whether that method should be included in the process of construction of searching thesauri. This paper presents also three steps for the construction of thesauri: acquisition, analysis, and presentation of concepts and terms. In [35], authors discuss how bibliometric methods can be applied to thesaurus construction. The paper presents semiautomatic and automatic thesaurus construction. Unlike our solution, it focuses on one subject area. The other methods for automatic build of thesauri and/or controlled vocabularies are presented in [9] and [11]. These solutions, unlike CoVoMe, do not have formally described steps.

There are a few approaches that are more formal [21,36]. In [36], nine steps to construct a thesaurus systematically is proposed. Unlike our solution, this proposal only focuses on one vocabulary for building CVs. The next formal approach is ISO 25964-1 [21] that explains how to construct thesaurus, how to display it, and how to manage its development. Unfortunately, this proposal only focuses on one vocabulary for building CVs. That proposal, unlike CoVoMe, describes process of building only one type of CV.

On the other hand, over the years, a considerable amount of research has been performed on user-centered approaches for the construction of thesauri and/or controlled vocabularies. In [32], the author focuses on the situational context that surrounds the user.

In [3], a thesaurus-based methodology is proposed for systematic ontological conceptualization in the manufacturing domain. The methodology has three main phases, namely, thesaurus development, thesaurus evaluation, and thesaurus conversion and it uses SKOS as the thesaurus representation formalism. That proposal, unlike CoVoMe, only focuses on one vocabulary for building CVs. Similar disadvantage can describe a methodology for a Thesauri Quality Assessment [34]. This proposal supports decision makers in selecting thesauri by exploiting an overall quality measure, but support only SKOS.

4.2 Methodologies for Ontology Development

In contrast to the construction of controlled vocabularies approaches, methodologies for ontology development are described more formally. They define steps to meet in the process of ontology development and determine how to document the process. Unfortunately, all the solutions below describe the process of creating an ontology and cannot be easily adapted as methodologies for building CVs.

METHONTOLOGY [14] is a construction methodology for building ontologies. In general, it provides a set of guidelines about how to carry out the activities identified in the ontology development process. It supports the techniques used in each activity, and the output produced by them. METHONTOLOGY consists of the identification of the ontology development process where the main activities are identified, a lifecycle based on evolving prototypes, and the methodology itself, which specifies the steps. Some steps in this methodology are similar to our proposal, e.g., specification can be comparable to step 1, and integration is similar to step 7 in CoVoMe.

On-To-Knowledge [40] is another methodology for building ontologies. It should be used by the knowledge management application because the methodology supports ontologies taking into account how the ontology will be used in further applications. Consequently, ontologies developed with On-To-Knowledge are dependent on the application.

Another methodology for ontology development is NeOn [39]. It supports, among others the reuse of ontologies as well as of non-ontological resources as part of the engineering process. This methodology also proposes detailed guidelines for executing its various activities. In contrast to our proposal, as well as to METHONTOLOGY and On-To-Knowledge that provide methodological guidance for ontology engineering, this methodology rather just recommends a variety of pathways for developing ontologies.

OD101 [33] is an iterative methodology that focuses on guidelines to formalize the subject domain by providing guidance on how to go from an informal representation to a logic-based one. It encompasses not only axiom choice, but also other aspects that affect that. A characteristic feature of this methodology is that, like our proposal, it is close to a vocabulary that can be used to construct an ontology. That proposal, unlike CoVoMe, is strongly connected to OWL. On the other hand, some steps of OD101 can be considered similar to CoVoMe steps, e.g., *define the classes, and the class hierarchy* step can be seen as similar to step 5 in our proposal.

Both NeOn and OD101, like CoVoMe, use Competency Questions [17] in the specification stage. This approach specifies what knowledge has to be entailed in the ontology and thus can be seen as a set of requirements on the content, as well as a way of scoping. We also use CQs in our methodology.

There are many different proposals that relate to the Rational Unified Process (RUP) [22–24]. The first approach [22], in addition to the RUP, is also related to traditional waterfall. The stages proposed by the methodology are based on the METHONTOLOGY. Incremental and Iterative Agile Methodology (IIAM) [23], which is the second proposal, unlike CoVoMe, is the domain-specific solution for the education field. Software Centric Innovative Methodology (SCIM) [24] has five ontology development workflows: requirements analysis, domain analysis, conceptual design, implementation and evaluation. Our proposal, like the above solutions, can integrate into RUP phases and disciplines.

Besides IIAM, there are other domain-specific methodologies, e.g. Yet Another Methodology for Ontology (YAMO) [13]. That methodology provides a set of ontology design guiding principles for building a large-scale faceted ontology for food.

5 Conclusions

In this paper, we have described a controlled vocabulary methodology for knowledge organization systems. We have listed the steps and activities in the CV development process. Our methodology has addressed the complex issues of defining concepts, concept schemas, semantic relations, mapping relations, labels, notation, and documentation. The advantages of CoVoMe are a direct consequence of its generality, including the support for different types of CVs and the possibility to use various vocabularies to create them. The proposed methodology can be used with different vocabularies for building CVs, as well as it flexibly supports different types of CVs.

As part of our future work, we will consider possibilities for enhancement by adding Notation3 rules that can help with evaluation. Furthermore, we intend to work on systematic monitoring of the adoption and use of CoVoMe in different areas, focusing on the problems that will emerge during the CVs creation process.

A Used Namespaces

Prefix	Namespace	Representation
`dcterms`	`http://purl.org/dc/terms/`	RDF
`gvp`	`http://vocab.getty.edu/ontology#`	RDF
`iso-thes`	`http://iso25964.org/`	XML
	`http://purl.org/iso25964/skos-thes#`	RDF
`mads`	`http://www.loc.gov/mads/v2`	XML
	`http://www.loc.gov/mads/rdf/v1#`	RDF
`skos`	`http://www.w3.org/2004/02/skos/core#`	RDF
`skos-xl`	`http://www.w3.org/2008/05/skos-xl#`	RDF

References

1. Alexiev, V., Isaac, A., Lindenthal, J.: On the composition of ISO 25964 hierarchical relations (BTG, BTP, BTI). Int. J. Digit. Libr. **17**(1), 39–48 (2015). https://doi.org/10.1007/s00799-015-0162-2
2. Altınel, B., Ganiz, M.C.: Semantic text classification: a survey of past and recent advances. Inf. Process. Manage. **54**(6), 1129–1153 (2018). https://doi.org/10.1016/j.ipm.2018.08.001
3. Ameri, F., Kulvatunyou, B., Ivezic, N., Kaikhah, K.: Ontological conceptualization based on the simple knowledge organization system (SKOS). J. Comput. Inf. Sci. Eng. (2014)
4. ANSI: American national standard guidelines for thesaurus structure, construction, and use (1980)
5. Bechhofer, S., Miles, A.: SKOS simple knowledge organization system reference. W3C recommendation, W3C, August 2009. https://www.w3.org/TR/2009/REC-skos-reference-20090818/
6. British Standards Institution: British standard guide to establishment and development of monolingual thesauri (1987)
7. Caracciolo, C., et al.: The AGROVOC linked dataset. Semant. Web **4**(3), 341–348 (2013). https://doi.org/10.3233/SW-130106
8. Chan, L.M.: Library of Congress subject headings: principles and application. ERIC (1995)
9. Chen, H., Lynch, K.J.: Automatic construction of networks of concepts characterizing document databases. IEEE Trans. Syst. Man Cybern. **22**(5), 885–902 (1992). https://doi.org/10.1109/21.179830
10. Coulter, N.: ACM's computing classification system reflects changing times. Commun. ACM **40**(12), 111–112 (1997). https://doi.org/10.1145/265563.265579
11. Crouch, C.J.: An approach to the automatic construction of global thesauri. Inf. Process. Manage. **26**(5), 629–640 (1990). https://doi.org/10.1016/0306-4573(90)90106-C
12. Deutsches Institut für Normung: Erstellung und weiterentwicklung von thesauri (1993)
13. Dutta, B., Chatterjee, U., Madalli, D.P.: YAMO: yet another methodology for large-scale faceted ontology construction. J. Knowl. Manage. (2015). https://doi.org/10.1108/JKM-10-2014-0439
14. Fernández-López, M., Gómez-Pérez, A., Juristo, N.: METHONTOLOGY: from ontological art towards ontological engineering. In: Engineering Workshop on Ontological Engineering (AAAI97) (1997)
15. Gandon, F.: Distributed Artificial Intelligence and Knowledge Management: ontologies and multi-agent systems for a corporate semantic web. Ph.D. thesis, Université Nice Sophia Antipolis (2002)
16. Gartner, R.: MODS: metadata object description schema. JISC Techwatch Rep. TSW 03–06 (2003)
17. Grüninger, M., Fox, M.S.: The role of competency questions in enterprise engineering. In: Rolstadås, A. (ed.) Benchmarking — Theory and Practice. IAICT, pp. 22–31. Springer, Boston (1995). https://doi.org/10.1007/978-0-387-34847-6_3
18. Heald, J.H.: The making of TEST thesaurus of engineering and scientific terms. Clearinghouse for Federal Scientific and Technical Information (1967)
19. Hodge, G.: Systems of Knowledge Organization for Digital Libraries: Beyond Traditional Authority Files. ERIC (2000)

20. International Organization for Standardization: Documentation-guidelines for the establishment and development of monolingual thesauri (1985)
21. International Organization for Standardization: Thesauri for information retrieval (2011)
22. John, M.S., Santhosh, R., Shah, N.: Proposal of an hybrid methodology for ontology development by extending the process models of software engineering. Int. J. Inf. Technol. Convergence Serv. **6**(1), 37–44 (2016). https://doi.org/10.5121/ijitcs.2016.6104
23. John, S., Shah, N., Smalov, L.: Incremental and iterative agile methodology (IIAM): hybrid approach for ontology design towards semantic web based educational systems development. Int. J. Knowl. Eng. **2**(1), 13–19 (2016). https://doi.org/10.18178/ijke.2016.2.1.044
24. John, S., Shah, N., Stewart, C.D., Samlov, L.: Software centric innovative methodology for ontology development. In: 9th International Conference on Knowledge Engineering and Ontology Development (KEOD-2017), pp. 139–146 (2017). https://doi.org/10.5220/0006482901390146
25. Krooks, D.A., Lancaster, F.W.: The evolution of guidelines for thesaurus construction. Libri (1993). https://doi.org/10.1515/libr.1993.43.4.326
26. Lipscomb, C.E.: Medical subject headings (MeSH). Bull. Med. Libr. Assoc. **88**(3), 265 (2000)
27. McIlwaine, I.C.: The universal decimal classification: some factors concerning its origins, development, and influence. J. Am. Soc. Inf. Sci. **48**(4), 331–339 (1997). https://doi.org/10.1002/(SICI)1097-4571(199704)48:4⟨331::AID-ASI6⟩3.0.CO;2-X
28. Mitchell, J.S., Beall, J., Matthews, W., New, G.: Dewey decimal classification. Encycl. Libr. Inf. Sci. (1996)
29. Needleman, M.: Standards update: some interesting XML standards. Serials Rev. **31**(1), 70–71 (2005). https://doi.org/10.1016/j.serrev.2004.11.012
30. Neubert, J.: Bringing the "thesaurus for economics" on to the web of linked data. LDOW **25964**, 102 (2009)
31. Nielsen, M.L.: The word association test in the methodology of thesaurus construction. Adv. Classif. Res. Online **8**(1), 41–57 (1997). https://doi.org/10.7152/acro.v8i1.12727
32. Nielsen, M.L.: A framework for work task based thesaurus design. J. Documentation (2001). https://doi.org/10.1108/EUM0000000007100
33. Noy, N.F., McGuinness, D.L., et al.: Ontology development 101: A guide to creating your first ontology (2001)
34. Quarati, A., Albertoni, R., De Martino, M.: Overall quality assessment of SKOS thesauri: an AHP-based approach. J. Inf. Sci. **43**(6), 816–834 (2017). https://doi.org/10.1177/0165551516671079
35. Schneider, J.W., Borlund, P.: Preliminary study of the potentiality of bibliometric methods for the construction of thesauri. In: Emerging Frameworks and Methods: Proceedings of the Fourth International Conference on Conceptions of Library and Information Science (CoLIS 4), Seattle, pp. 151–165 (2002)
36. Shearer, J.R.: A practical exercise in building a thesaurus. Cataloging Classif. Q. **37**(3–4), 35–56 (2004). https://doi.org/10.1300/J104v37n03_0
37. Soergel, D.: The art and architecture thesaurus (AAT): a critical appraisal. Vis. Resour. **10**(4), 369–400 (1995). https://doi.org/10.1080/01973762.1995.9658306
38. Souza, R.R., Tudhope, D., Almeida, M.B.: Towards a taxonomy of KOS: dimensions for classifying knowledge organization systems. KO Knowl. Organization **39**(3), 179–192 (2012). https://doi.org/10.5771/0943-7444-2012-3-179

39. Suárez-Figueroa, M.C., Gómez-Pérez, A., Fernández-López, M.: The NeOn methodology for ontology engineering. In: Suárez-Figueroa, M.C., Gómez-Pérez, A., Motta, E., Gangemi, A. (eds.) Ontology Engineering in a Networked World, pp. 9–34. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-24794-1_2
40. Sure, Y., Staab, S., Studer, R.: On-to-knowledge methodology (OTKM). In: Staab, S., Studer, R. (eds.) Handbook on Ontologies. INFOSYS. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24750-0_6