An Analysis on the Use of Knowledge Organization Systems in the Process of Requirements Engineering

Priscila Basto Fagundes* and Douglas Dyllon Jeronimo de Macedo**

Department of Information Science, Federal University of Santa Catarina,
Campus Professor João David Ferreira Lima Bloco B sala 105
Florianópolis, Santa Catarina, Brasil 88.040-900
*<priscila.fagundes@prof.usj.edu.br>, **<douglas.macedo@ufsc.br>

Priscila Basto Fagundes is a Ph.D. student in Information Science at the Federal University of Santa Catarina (UFSC), Master in Computer Science at UFSC (2005), and Bachelor in Systems Analysis at the Catholic University of Pelotas (1998). Since 2004, she has been teaching undergraduate and graduate courses. Currently a member of the ITI-RG research group: Intelligence, Technology, and Information linked to the Information Science Department at UFSC and external researcher at the Institute of Systems and Computer Engineering, Technology and Science (INESC TEC) in the city of Porto in Portugal. Her areas of interest are Information and Communication Technology, Information Management, Information Quality, Software Engineering, Requirements Engineering, and Knowledge Organization Systems.

Douglas Dyllon Jeronimo de Macedo is Adjunct Professor at the Department of Information Science (CIN), Federal University of Santa Catarina (UFSC). He holds a doctorate in Knowledge Engineering (EGC) from the Federal University of Santa Catarina (UFSC). During his doctorate period, he worked as a visiting researcher at The University of Western Ontario (UWO), in London, Ontario, Canada. He works as an associate researcher at the National Institute of Digital Convergence (INCoD), working at the Telemedicine Laboratory (LabTelemed) and the Distributed Systems Research Laboratory (LaPeSD) at UFSC. He also serves as a Permanent Member of the Graduate Program in Computer Science (PPGCC / UFSC) and the Graduate Program in Information Science (PGCIN / UFSC). He is the coordinator of the Data Science and Engineering Laboratory (LECID) at UFSC.

Fagundes, Priscila Basto and Douglas Dyllon Jeronimo de Macedo. 2022. "An Analysis On The Use Of Knowledge Organization Systems In The Process Of Requirements Engineering." *Knowledge Organization* 49(6): 411-422. 54 references. DOI:10.5771/0943-7444-2022-6-411.

Abstract: Some of the fundamental activities of the software development process are related to the discipline of Requirements Engineering. Their objectives are to discover, analyze, document, and verify the system's requirements. The requirements are the conditions or capabilities that software needs to have or fulfill to meet its users' needs, and problems in its identification can mean the failure of a software project. This study is part of the research that is being developed to propose a model based on Knowledge Organization Systems to be used in the Requirements engineering process. This article aims to present the results of an analysis on a set of Knowledge Organization Systems to identify whether they are likely to be applied in the Requirements engineering process and identify at which stage of this process each one of them can be implemented. The Knowledge Organization Systems analyzed were the authority files, gazetteers, glossaries, subject headings, classification systems, thesauri, semantic networks, and ontologies. Based on the results obtained, it was possible to conclude that the Knowledge Organization Systems analyzed can be used in the Requirements engineering process and, consequently, contribute to increasing the software requirements' quality.

Received: 17 March 2021; Accepted 7 June 2022

Keywords: requirements engineering, modelling, knowledge organization tools

1.0 Introduction

Increasingly, organizations depend on the support provided by software to provide information, which has a fundamental role not only in the decisions to be made but also in the activities developed by its users, which leads to the need for this information to be available, reliable and useful (Gharib et al. 2018; Morales-Ramirez et al. 2015). According to Sommer-

https://doi.org/10.5771/0943-7444-2022-6-411 - am 24.01.2026, 08:25:44. https://www.inlibra.com/de/agb - Open Access -

ville (2016) and Pressman (2014), the construction of quality systems that meet their users' information needs is strongly related to the process used for its development.

The area of computing called software engineering aims to present concepts, techniques, and tools to be applied during this process, which include the steps of specification, design, coding, validation, and evolution of the software. During the specification stage of software engineering are defined the features of, and restrictions to, the operation of the system. And this stage counts on the requirements engineering area to support the discovery, analysis, documentation, and validation of software requirements.

Dal Forno et al. (2017) and Sweis (2015) identified that the problems more frquently mentioned about critical factors that contribute to the failure of projects that involve systems development, is directly related to activities of the requirements engineering, since errors committed in this step, if they are not corrected, extend throughout the development process and after checking each error, all previous phases need to be redone. The article by Fernandez et al. (2017) presents a survey with the participation of 228 software development companies located in 10 countries to identify, based on the experience of requirements engineering professionals, which problems harm this process; according to the authors, the main difficulties identified were: incomplete or hidden requirements; communication failures between the project team and the client; changing objectives, business processes or requirements; and abstract requirements.

What is noticeable is that the same problems reported in Fernandez et al. (2017) are cited in previous studies as in Faulk (1997), Hall et al. (2002), Liu et al. (2010), Nuseibeh and Easterbrook (2000), Przybyłek (2014) and Solemon et al. (2009). That is, despite the efforts of researchers in the software development area to propose techniques and tools that help in carrying out their activities, the problems that contribute to the failure of software projects and users' dissatisfaction are still the same as those of 23 years ago.

Since the focus of requirements engineering is to identify and transform information into software requirements, we considered relevant a reflection about using approaches from other areas of knowledge to assist in this process. In this way, we decided to research outside the borders of computing, techniques that could contribute to the increase of the quality of the requirements and consequently with the rise of the users' satisfaction; e based on the research carried out, we identified opportunities to use some of the types of knowledge organization systems (KOS) to assist requirements engineering in the execution of its activities.

This article aims to present the results of an analysis carried out on a set of knowledge organization systems to verify if these systems could be used in the context of requirements engineering and identify in which activity or activities in this process they could be implemented. The knowledge organization systems analyzed were proposed by Hodge (2000): authority files, gazetteers, glossaries, subject headings, classification systems, thesauri, semantic networks, and ontologies. The analysis was made considering each mentioned KOS's characteristics and objectives, allowing identifying which stages in the Requirements engineering process they could be applied. And according to the results obtained so far, it was possible to conclude that, except for gazetteers, the KOS studied can be used to support the requirements engineering adding value to its process.

As for contributions to the field, we expect this study to encourage new research involving knowledge organization systems and requirements engineering areas to improve the process related to software requirements. This includes contributions to software development companies that will be able to use knowledge organization systems during the requirements engineering process, increasing the quality of the developed software and, finally, contributes with the requirements engineers, as an opportunity to know and use techniques outside the frontiers of computing.

Section 2.0 presents the concepts and definitions of requirements engineering and the knowledge organization systems we choose to analyze. Section 3.0 presents the methodological procedures used to achieve this study's goals and explains how the analysis was carried out. Section 4.0 presents studies already carried out involving some of the knowledge organization systems and the requirements engineering. Section 5.0 presents the results of analyzing, and section 6.0 presents the conclusions of this paper.

2.0 Theoretical background

2.1 The process of requirements engineering

The area known by the term requirements engineering emerged as an independent field of study in the early 1990s, leveraged by creating the conference called *International Requirements Engineering Conference – IEEE* and the journal *Requirements Engineering*, published by Springer Verlag (Nuseibeh and Easterbrook 2000). Requirements engineering is considered a subarea of software engineering, and it belongs to the specification stage in the development process, in which the requirements engineer or the person responsible for this activity seeks to understand what are the real needs of users concerning resolution or dissolving the problem that needs to be solved. Requirements engineering can be described as a set of activities aimed at discovering, analyzing, documenting, and validating the requirements and documents generated (Kotonya and Sommerville 1998).

Regarding the understanding of users' needs, requirements are considered a critical factor in software projects. If

https://doi.org/10.5771/0943-7444-2022-6-411 - am 24.01.2026, 08:25:44. https://www.inlibra.com/de/agb - Open Access -

they are not well defined, they directly impact the success of any product that will be delivered. Also, they are used to measure the size, complexity, and consequently, the cost of software, and when problems with requirements identification are detected late, there is a high cost to correcting them.

According to Leffingwell and Widrig (2003), users' needs represent a business, personal or operational problem that justifies the need to develop or acquire a new information system. Kruchten (2003) presents that a software requirement is a condition or a capacity that software needs to have or perform to meet its users' needs. Sommerville (2016) expands this meaning, defining a system's requirements as the descriptions of what the system should do, the services it offers, and the restrictions on its operation. These requirements reflect the user's needs for a system that serves a specific purpose, such as controlling a device, placing an order, or finding information.

Goguen (1996) and Jackson (1995) share this viewpoint, affirming requirements are information, and all information has a context, and that it is the contexts that determine the nature of the requirements. According to the authors, it is necessary to consider how information is produced and used and not just represented. The situations that determine the nature of the requirements are not only technical since they significantly involve the social context in which the users of the system are inserted, which ends up resulting in requirements arising from different points of view, a fact that requires special attention to identify them effectively, allowing an abstract representation of the system (Rocha and Vasconcelos 2004).

In general, the requirements are defined as specifications of the system's objectives and behaviors, together with the restrictions imposed on its use. According to Sommerville and Sawyer (1997), a software requirement can describe a user's need (e.g. "the system must calculate the financial burden on employees"), public ownership of the system (e.g. "the system must control all access using a login and password"), a restriction on its operationalization (e.g. "the sensor must capture information at a speed of 10 times per second") or a restriction on its development (e.g. "The system must use the MySQL database management system").

The requirements engineering process is related to discovering the contextual reasons for the system's existence, the functionalities necessary to meet these reasons, and the existing restrictions for the execution of these functionalities (Lamsweerde 2000). Its objective is elaborating a complete, unambiguous, consistent, and correct specification of the requirements, making these serve as the basis for the other stages of software engineering.

To Kotonya and Sommerville (1998), requirements engineering is the area that includes the activities involved in the process of discovering, documenting, and maintaining a set of software requirements, the term "engineering" being related to the use of systematic and that must ensure that the system requirements are complete, consistent and relevant. According to Nuseibeh and Easterbrook (2000), requirements engineering is composed of a set of activities that focus on identifying and communicating the objectives of a system and the context in which it will be used, making an intersection between the needs of users and the potential and opportunities offered by technology. A more recent definition is given by Dick et al. (2017) that define requirements engineering as a subset of systems engineering concerned with the discovery, development, traceability, analysis, quality, communication, and management of the requirements that define the system in successive levels of abstraction.

Software development processes are considered complex. The definition of which and how activities should be implemented depends on factors, such as the type of software that will be developed, the organizational culture, or the resources involved. What authors like Pressman (2014), Sommerville (2016) and Robertson and Robertson (2013) affirm is that, although there is no ideal process, it is essential that one exists and is defined as aiming at the development of quality software.

It is possible to find in the literature different proposals related to the activities that are part of the requirements engineering process and for the analysis presented in this article, the activities proposed by Pressman (2014), which suggests that the requirements engineering should provide the mechanism appropriate to understand what the user wants, analyzing the needs, evaluating the feasibility, negotiating a reasonable solution, specifying the solution without ambiguities, validating the specification and managing the needs as they are transformed into an operating system.

The process proposed in Pressman (2014) includes the seven activities described below:

- Conception: aims to establish a basic understanding of the problem, identifying those involved, the nature of the desired solution, and the effectiveness of preliminary communication and collaboration between users and the project team.
- Elicitation: identifies the objectives, functionalities, and needs requiring by users.
- Elaboration: refines the information obtained during the design and develops a technical model containing the software's functions, characteristics, and restrictions.
- Negotiation: It carries out a negotiation between those involved in the project, aiming to prioritize the software requirements.
- Specification: formalizes/documents the requirements to support subsequent activities in the development process.

https://doi.org/10.5771/0943-7444-2022-6-411 - am 24.01.2026, 08:25:44. https://www.inlibra.com/de/agb - Open Access -

- Validation: evaluates and validates the generated artifacts, where software engineers, customers, users, and other stakeholders participate in the validation.
- Management: performs requirements management, defining individual identifications and relationships, allowing their traceability, and facilitating changes when necessary.

According to Bourque and Fairley (2014), an ideal environment for the requirements engineering activities' performance is when the users and those responsible for the project's requirements work together in the same team. In this case, it becomes just a matter of conduction of the interactions. However, in some cases, the reality is quite different, and in most cases, the interactions do not occur as they should, harming the process as a whole. Factors such as lack of communication, problems in the exchange of information, and the sharing of knowledge between those involved in the process negatively impact the results of the activities carried out.

Requirements engineering's difficulties can be analyzed based on two major groups: accidental difficulties and essential difficulties. Accidental difficulties are those that originate from a lack of understanding about what really needs to be developed. The following can be highlighted: little effort spent on activities related to the identification of information with users, the lack of documentation on the requirements obtained, the failure to carry out the checks and validations in the identified requirements, incorrect specifications of the requirements, the lack of management of the elicited requirements and the tendency to carry out the other stages of software engineering without the requirements engineering having been effectively carried out. Essential difficulties, on the other hand, are those that are beyond the control of requirements engineering, among them, the user's difficulty in knowing effectively what he expects from the software, communication problems between the user and the requirements engineers, and also the changes that requirements suffer throughout the process (Faulk 1997; Przybyłek 2014).

The accidental difficulties can be considered easier to overcome if compared to essential difficulties since adopting a systematic process that guides the Requirements engineering activities tends to solve, or at least significantly minimize, problems in this category. The essential difficulties are more difficult to overcome, as they are part of the software's nature, which is abstract, malleable, and complex, and are also beyond the development team's control.

2.2 Knowledge organization systems

According to Carlan and Medeiros (2011), among the tools to organizing and retrieving information, there are those known as knowledge organization systems. For the authors, KOS aim at terminological standardization to facilitate indexing and guide users. In terms of their structure, they can vary from a simple to a multidimensional scheme. At the same time, their functions include eliminating ambiguity, control of synonyms or equivalents, and establishing semantic relationships between concepts.

For Hjørland (2003), knowledge organization systems organize concepts and their semantic relations. According to Stock (2010), a knowledge organization system is composed of semantic concepts and relationships that represent a domain of knowledge in order to support the recovery process. In Gödert et al. (2014), the expression "knowledge organization system" means "knowledge structure", formed by related concepts and built to provide support for indexing and retrieving information.

According to Hodge (2000), knowledge organization systems include different types of tools used to organize information and enable knowledge management. The author proposes that knowledge organization systems be classified into three general categories:

1. term lists, which include lists of terms often accompanied by definitions;
2. classifications and categories, which highlight the creation of sets of subjects;
3. relationship lists, which highlight the connections between terms and concepts.

Table 1 summarizes the tools suggested by Hodge (2000) to be used in each of the categories proposed by him.

Category	KOS
Lists of terms	Authority Files
	Glossaries
	Gazetteer
Classifications and categories	Subject headings
	Classification systems, taxonomies, and categorization systems
Relationship lists	Thesaurus
	Semantic networks
	Ontology

Table 1. Types of knowledge organization systems according to Hodge (2000).

We decided that the knowledge organization systems proposed by Hodge (2000) would be the most suitable to form part of the analysis presented in this article because they are

https://doi.org/10.5771/0943-7444-2022-6-411 - am 24.01.2026, 08:25:44. https://www.inlibra.com/de/agb - Open Access -

used by authors like Shiri and Molberg (2005) and McCulloch and MacGregor (2008) and are also adopted by the Networked Knowledge Organization Systems and Services (NKOS).

For Harpring (2010, 21), an authority file "is a set of established names or headings and cross references to the preferred form from variant or alternate forms" and can refer to both a specific methodology and a controlled vocabulary in particular. According to Grings (2015), authority files include an alphabetical index, and within each index are cross-references that relate headings. A list is a form of record that helps in cataloging to retrieve information and includes the standardized form of names, subjects, and subdivisions, aiming to standardize access points. The elaboration of a catalog of authorities is considered fundamental for the organization and retrieval of information in any documentary unit. And it is considered a derivative and auxiliary tool of the bibliographic catalog, whose basic function is to establish standardized access points (authorities) that will serve users as search keys, guaranteeing the reliable and effective location of information.

An authority file is a database composed of authority records that contain: authorized entries, unauthorized and related entries (cross-references), bibliographic sources consulted, and notes that clarify the choice of entry and the decisions made by the person responsible for cataloging (Hodge 2000; Tillett 2004). The types of entries available in an authority file are usually names (personnel, entity, event, geographic), uniform titles, series and subjects. This type of catalog has the following functions: (i) to assist the user in locating documents of interest, organized in a single entry; (ii) increase accuracy in retrieving the information made by the user in the catalog; (iii) increase confidence in the information provided by the catalog to users; (iv) increase cataloging efficiency and maximize resources; and (v) assist navigation between records.

Glossaries contain a set of terms and their meanings used in a given area of knowledge. It is usual that the glossaries be presented in alphabetical order and include explanations of relevant concepts from a specific field of study or action (Hodge 2000). Glossaries are considered a relatively simple tool to structure and use since it does not require complex techniques for its development and compression. However, it is essential to know what information is relevant to be included in the glossary and not leave any important terms out of the list.

A gazetteer is a geographic dictionary, digital or otherwise, that associates place names with geographic coordinates that contain triples: place names (N), entities (feature types) (T), and geometric representations with coordinate geographic footprints (F). In general, gazetteers have two functions, one that maps names to entities and the other that maps names to resource types. Resource types are mainly organized in semi-formal thesauri with natural language descriptions (Keßler 2009).

Gazetteers are often organized using classification schemes or subject headings, and the vocabulary used for resource types can vary between geographical dictionaries and may include terms such as "airport", "port", or "train station". According to Hodge (2000), with this type of knowledge organization system, the names of places in a library catalog or a bibliographic database can have a set of assigned characteristics. Therefore, the way information must be presented to the user is designed to allow him to distinguish the locations. An example of using this type of KOS can be in ecology, environmental science, and even public health, where users can access the system through text mode or geographic mode, depending on the type of information needed. Presenting the results on a map allows users to make new associations and analyze the results more easily.

The lists of subject headings were proposed to provide instruments for indexing the subjects in documents, which would be registered in catalogs to create the alphabetical catalog of subjects. The headings' purpose was to help the libraries of general collections to understand knowledge as a fragmentable universe, in disciplines, by providing a set of controlled terms to represent the subjects of the items in a collection.

According to Harpring (2010), subject headings are uniform words or phrases intended for use with books, articles, or other documents, to describe their subject or topic, and group them based on their subjects' affinity. The subject headers most commonly used in libraries in the United States are the Library of Congress Subject Headings (LCSH), which form a comprehensive list of preferred terms or strings, usually with cross-references. Another well-known set of subject headings is Medical Subject Headings (MeSH), which is used for indexing journal articles and books on medical science. For example, in LCSH and MeSH, parts of a composite header can be stored in subfields of the Machine Readable Cataloging (MARC) format, separated to allow variations in the displays as desired (Harpring 2010). The lists of subject headings can be extensive and cover a wide range of subjects. However, they are considered superficial by some authors who believe their hierarchy is limited (Hodge 2000).

Classification systems are artificial systems that allow an easier and more effective representation of certain documentary content, aimed at the manual or automatic retrieval of information requested by the user. The classification systems most cited in the literature are the Dewey Decimal Classification (DDC), the Universal Decimal Classification (UDC), the Library of Congress Classification (LCC), and faceted classification theory (FCT). However, due to the specificity of the application of the DDC, UDC, and LCC systems in the scope of libraries, only FCT will be

https://doi.org/10.5771/0943-7444-2022-6-411 - am 24.01.2026, 08:25:44. https://www.inlibra.com/de/agb - Open Access -

considered in this study since it proves to be applicable also outside the context of librarianship.

Developed by the Indian librarian and mathematician Shiyali Ramamrita Ranganathan in the 1930s, faceted classification theory has the principle of dividing a subject between its different aspects, in groups of classes brought together by the same principle of division that function as a synthetic, analytical scheme. FCT involves two distinct processes: the analysis of the subject in facets and the synthesis of the elements that constitute the subject in question, which can be applied to any knowledge area (Broughton 2006).

According to Tristão et al. (2004), the classification process by facets starts from an analysis of the theme, dividing it into its constituent parts and decomposing the most complex elements (subjects) into simple concepts (basic concepts or facets), seeking to synthesize, condense and examine each of these parts, and then join them according to the characteristics of the document to be described and represented. In faceted systems, the division is cyclical, that is, a given subject is divided into subclasses until the possible variations are exhausted. This classification system consists of three work planes: the plane of ideas, the verbal plane, and the notational plane.

Concerning taxonomy, it can be defined as a classification system that supports access to information, allowing to classify, allocate, retrieve and communicate information in a system in a logical manner. According to Vital and Café (2011), a taxonomy organizes information from the most generic to the most specific, using the hierarchical or gender-species relationship between terms. Aiming at the effective retrieval of information, taxonomies are directed to organize information in specific environments in an orderly knowledge creation process. Basirati et al. (2015) indicate that taxonomy is used in different areas, including computing, to structure the information presented to software users since it is considered a relevant tool for retrieving information through navigation.

Thesauri consist of specialized language used for documentary purposes, in which the linguistic elements that compose it are syntactically and semantically related. Their purpose is to control the terms used in indexing using an instrument that translates the natural language of the authors, users, and indexers into a more controlled language and to standardize indexing in a cooperative network, limiting the number of terms to what is actually necessary, assisting in the task of information retrieval (Garshol 2004; Gilchrist 2003). According to the authors, for the development of a standardized and consensual language, aspects involving description, organization and information retrieval are necessary.

Ramalho et al. (2008) point out that there is a tendency for the development of thesauri to be based on digital environments and that they have limitations in the exploration of certain informational content, since, even though they allow the construction of a flexible structure of relationships between concepts, do not have sufficient expressiveness to represent rivher semantic relations between them.

Semantic networks are used to visually represent knowledge through oriented graphs, usually connected and cyclical, whose focus is on the categories of objects and the relationships between them. In this type of knowledge organization system, knowledge is represented by a set of nodes (nodes) and a set of arcs (links). The nodes represent concepts through nouns, adjectives, pronouns, or proper names, and the arcs represent the relationships between the concepts through transitive verbs or prepositions (Sowa 1991).

According to Harpring (2010), semantic relations are sometimes derived from vocabularies. They can be used to map terms from one or more vocabularies controlled according to an organizational structure or conceptual scheme. Relationships can range from a simple hierarchical structure with generic relationships to a more complex set of carefully defined relationships. Relationships can be categorized to indicate the degree of proximity between terms, for example, exact synonyms, near-synonyms, closely related terms, loosely associated terms, and antonyms.

Ontologies are structured by a list of concepts or entities within a specific domain, structured hierarchically through semantic relations formally explained in a computerized médium (Lima and Maculan 2017). Vital and Café (2011) claim that an ontology is formed by four essential components: the classes that are organized in a taxonomy; the relationships between them that represent the type of interaction between the concepts of a domain; the axioms that are used to define sentences (always true); and the instances, which are representations of the data itself.

In computing, the term ontology began to be used in the early 1990s in projects to organize large knowledge bases. According to Kilov and Sack (2009), for computer professionals, an ontology is a document or file that formally defines the relationships between terms and concepts, similar to the thesaurus used to define controlled vocabularies. For this area, the objective of building an ontology is to supply the need for a shared vocabulary for the exchange of information between members of a community, whether they are human beings or intelligent agents, serving as a basis to ensure unambiguous communication. In the area of software engineering, can be a variety of ontologies to be used are identified, which are classified as generic ontologies for software engineering, which aim to model the complete body of knowledge in the area, and specific ontologies which seek to conceptualize only a subdomain of this discipline (Ruy et al. 2016).

https://doi.org/10.5771/0943-7444-2022-6-411 - am 24.01.2026, 08:25:44. https://www.inlibra.com/de/agb - Open Access -

3.0 Methodology

Regarding the methodological procedures used to achieve the objectives of this article, exploratory bibliographic research was carried out to investigate concepts related to requirements engineering and knowledge organization systems. Lucio et al. (2013) clarify that the use of an exploratory study occurs when the objective is to examine a specific topic that has been little or never previously studied, providing a greater degree of familiarity with the phenomena involved. The study was carried out based on a survey and reading of theoretical references published in written and electronic media, such as books, scientific articles, and web pages. This investigation is considered basic research, as it aims to generate new knowledge that is useful for the advancement of science through the development of other research that will continue and complement it.

To achieve the objectives defined for this article, we conducted an analysis of each goal of the KOS, and we verified whether they displayed any adherence to the goals of each of the requirements engineering activities proposed in Presmann (2014). Based on this analysis, we identified at which stage or in which stages of the process each of the KOS can be implemented.

4.0 Related works

To establish this study's originality and to identify research involving knowledge organization systems assistance in the requirements engineering process, we searched for papers with this theme at Google Scholar in April 2019. We used the following terms "requirements engineering" AND "knowledge organization systems" and defined them as inclusion criteria for studies published between 2000 and 2018.

The result showed 27 publications. We read all the texts and found that none of them specifically addressed the use of knowledge organization systems to assist the requirements engineering process. We decided to carry out consultations in the same database because of the result obtained, but now involving terms related to each of the KOS analyzed in this article: lists of authorities, subject headings, glossaries, thesaurus, taxonomies, semantic networks, and ontologies, and requirements engineering. Some of the studies considered relevant are presented below.

Basirati et al. (2015) propose a taxonomy to assist in controlling changes in requirements. This taxonomy considers documents that contain information about software requirements. Another study with a similar purpose is the taxonomy established by Briand et al. (2003). This taxonomy is based on changes in the UML models and is used to identify changes in the different versions of the models and is intended to assist in determining the impact of these changes.

Ahmed et al. (2016) propose a Semantic Network Model model (SNM) to assist the requirements management process. The vertices and edges represent information about the requirements models and their relationships. The proposed model aims to help those responsible for the requirements through a semiautomatic standardization process, eliminating redundant information, avoiding excessive specifications, and allowing a general and simplified view of the entire system. With a focus on maintaining traceability between requirements, Mahmood et al. (2015) suggest an algorithm to find similarity between requirements and automate the retrieval of traceability links using the semantic network concepts, the DBpedia knowledge base, and the dictionary multilingual Bablenet 2.5.

Among the knowledge organization systems analyzed in this article, ontologies are the ones that present themselves as the most used by requirements engineering. Nardi and Falbo (2006) propose an ontology of basic concepts about requirements to be used as a basis for building tools to support the requirements engineering process. Assawamekin et al. (2010) present a framework called Multiperspective Requirements Traceability (MUPRET) that uses an ontology based on knowledge management for the automatic generation of relationship links to determine the traceability between requirements. Sharma and Ingle (2011) propose a framework called Ontology Aided Requirements Engineering (OntoAidedRE), which aims to support the definition of categories for the requirements to elicit, represent and analyze the diversity of factors associated with the requirements engineering process.

Sitthithanasakul and Choosri (2016) present a proposal for using ontologies to support requirements engineering in projects that use the Agile development approach. In this context, the ontology is treated as a formal tool to extract tacit knowledge and represent it visually. The authors suggest that using ontologies as a means of communication between stakeholders can improve the understanding of requirements. The proposed ontology aims to transform a list of requirements into a visual diagram to be more understandable by both the Agile software development team and the client/stakeholder.

We did not identify research proposing the use of lists of authorities, geographic dictionaries, subject headings, or thesauri in the requirements engineering process. We believe it is relevant to conclude this section, clarifying that the fact that studies involving some of the knowledge organization systems discussed here have been identified does not invalidate the analysis presented in this article. On the contrary, the identification of such studies confirms that these KOS can contribute to the requirements engineering process, and the proposal of a model that presents them in a single structure can enable those responsible for the requirements to decide which of them is more appropriate to use in each project.

https://doi.org/10.5771/0943-7444-2022-6-411 - am 24.01.2026, 08:25:44. https://www.inlibra.com/de/agb - Open Access -

5.0 Results and discussion

Considering that the requirements engineering process has information as to its main object and that information management comprises a set of linked activities related to the entire information cycle, we believe that knowledge organization systems can contribute with requirements engineering, mainly concerning the use of complementary approaches in understanding information needs and processing information for software development.

The analysis that will be presented below was based on the objectives of each of the techniques proposed in Hodge (2000) for the construction of knowledge organization systems, and the objectives of the activities of requirements engineering proposed in Presmann (2014). This analysis aimed to identify in which requirements engineering activity the KOS can be implemented, which will make it possible in the future to propose the model for the improvement of the requirements engineering processes using the knowledge organization systems. Figure 1 presents a structure contemplating the requirements engineering process's activities, the types of knowledge organization systems that can be used in each of them, and the expected results after their implementation.

Considering that the elaboration of an authority file is considered essential for the organization and retrieval of information in any documentary unit (Grings 2015), we believe that this KOS can collaborate with the implementation of mechanisms that assist the specification and management activities of the requirements, especially concerning the organization and management of the artifacts containing the requirements, as access to these documents can be established from the definition of standardized access points, such as those responsible for preparing the documents.

Glossaries are considered relatively simple tools, both to be used and to be developed, so they can be developed during the specification activity as part of the requirements documentation, assisting the development team and users in understanding the generated artifacts and information contained in them, as well as in communication between stakeholders.

During the specification activity, different types of documents are involved which contain information about the software requirements. For this reason, subject headings were related to this activity and also the requirements management activity. We believe that subject headings can assist in the organization of documents prepared during the re-

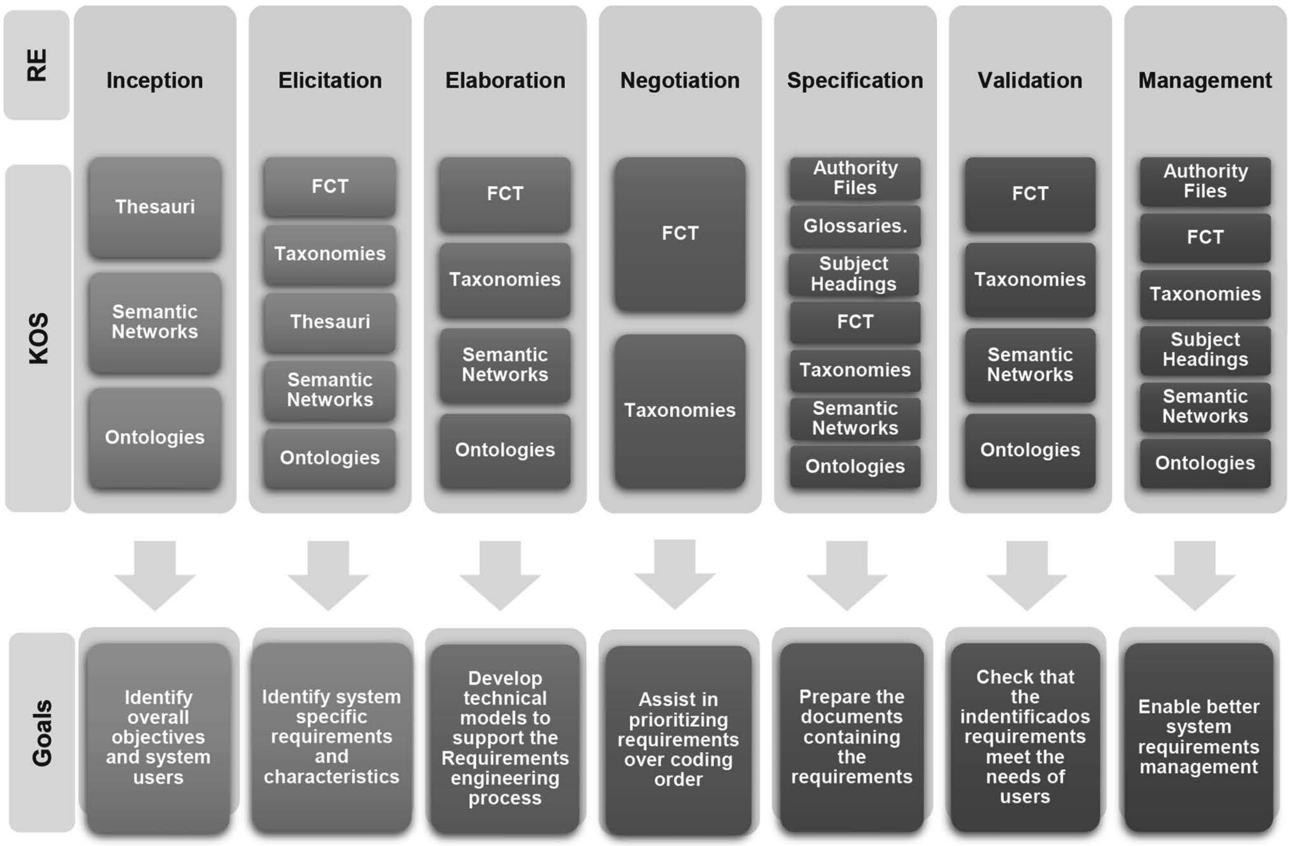

Figure 1. Analysis of KOS types and requirements engineering activities.

https://doi.org/10.5771/0943-7444-2022-6-411 - am 24.01.2026, 08:25:44. https://www.inlibra.com/de/agb - Open Access -

quirements engineering process and consequently can allow for better management of requirements.

Faceted classification theory and taxonomies can be applied during the activities of:

- elicitation of requirements since the action of classifying something requires an effort to understand what is being classified, which leads to a better understanding of the system requirements;
- elaboration and specification, since the classification systems generate models and structures to visually represent the classification that is being carried out, models that can be incorporated into the system documentation;
- negotiation, since the development team can use the groupings generated by the classifications to modularize the systems and, thus, prioritize the requirements based on the modules resulting from this group;
- validation, because the models and documents generated can serve as artifacts for the validation of requirements with users since they will contain structural information about the requirements; and
- management, since the classification systems, support access to information, allowing its classification, retrieval, communication, and organization, thus helping in the information management process.

Considering that thesauri are formed by a set of terms about a specific area of knowledge with their respective meanings and relationships, we believe that the development team could use this type of KOS during inception activities as an instrument to assist in understanding technical terms specific to the domain of the system, and during the activity of elicitation of the requirements for the definition of the relations between the system's functionalities.

Used to represent knowledge through visual models, the semantic networks could be used in the activities of:

- conception and elicitation, making it possible, through the relationships established, to understand the concepts, both at a more comprehensive level, helping to understand the general aspects, as well as at a more specific level, allowing the identification of data that needs to be stored and manipulated by the system;
- specification, in which the generated model can be used as an artifact belonging to the system documentation;
- validation, as it is considered a relatively simple model to be interpreted, the graph generated can be used by the development team to validate whether the identified requirements are in accordance with the information provided by users or domain experts, as well as to verify that it meets the requirements your expectations for the software; and
- management, since the models will be useful to assist in the definition of the storage structure of this information – which may be in a physical or digital medium – and of the dependencies between the requirements, helping in the maintenance process of these, as well as facilitating their traceability and recovery when needed.

Because they have characteristics and objectives similar to semantic networks, ontologies are candidates to assist the same set of activities as semantic networks in requirements engineering: design, elicitation, elaboration, specification, validation, and management. Ontologies can be used to provide a common and shared understanding of the domain, an understanding that can be communicated to people and to automated systems. The creation of an ontology of the system domain can also help define and establish the relationships between the generated requirements artifacts, assisting in the collaborative work among those involved in the project, as they create a unified structure of knowledge that represents the system.

No application possibilities for gazetteers were identified during the requirements engineering activities. As it is a tool for the construction of knowledge organization systems that seeks to specifically associate place names with geographic coordinates, it is understood that, due to this restriction, this type of KOS is outside the context of software development.

As can be seen, it was identified that, except for gazetteers, the other KOS selected to be part of this analysis presented relationships with at least one of the activities of the requirements engineering process since they expose in their objectives points of adherence to the objectives activities that are part of this process.

6.0 Conclusions

This paper seeks to identify possibilities for collaboration between the areas of information science and computer science concerning the use of knowledge organization systems to assist the development of activities belonging to the requirements engineering process, increasing the quality of the identified requirements and, consequently, of the developed software.

Although proposals are found in the literature with the purpose of assisting requirements engineering in gathering information for the construction of systems that are considered useful for its users, it is still possible to observe opportunities for improvement in this process. It should be noted that most studies with such an objective are developed within the computing area and do not take into account concepts that involve the nature of the information that will be transformed into the requirements, which is perfectly understandable, since the information in its more original sense is not the essence of the mentioned area.

https://doi.org/10.5771/0943-7444-2022-6-411 - am 24.01.2026, 08:25:44. https://www.inlibra.com/de/agb - Open Access -

The results of the analysis between the objectives of KOS and the activities of requirements engineering will be used for the proposal of a model that contemplates the KOS proposed in Hodge (2000) to be used in the context of the requirements engineering, being that the next stage of this research consists in the application of this set of KOS in a project that is being developed in a company in the south of Brazil. Knowledge organization systems will be applied and evaluated considering the following aspects: complexity, satisfaction, resources involved, and adaptability.

However, we believe that the results obtained so far and presented in this article are relevant and can help the scientific community to the extent that they can use them for the development of new research, including information science, requirements engineering and knowledge organization systems, such as the software development industry that can count on alternative techniques that its teams can use to identify requirements during the development of software projects.

References

Ahmed, Kushal, Lian Wen, Abdul Sattar and Reza Farid. 2016. "Semantic Network Model: A Reasoning Engine for Software Requirements." In *Proceedings of the IEEE International Conference on Engineering of Complex Computer Systems, (ICECCS) Gold Coast*, *Australia*. Institute of Electrical and Electronics Engineers, 194–7. doi:10.1109/ICECCS.2015.31

Assawamekin, Namfon, Thanwadee Sunetnanta, and Charnyote Pluempitiwiriyawej. 2010. "Ontology-Based Multiperspective Requirements Traceability Framework." *Knowledge and Information Systems* 25: 493–522.

Basirati, Mohammad R., Henning Femmer, Sebastian Eder, Martin Fritzsche and Alexander Widera. 2015. "Understanding Changes in Use Cases: A Case Study." In *IEEE 23rd International Requirements engineering Conference Ottawa, Canada*. IEEE, 352–61. doi:10.1109/RE.2015.7320452

Bourque, Pierre and Richard E. Fairley. 2014. *Guide to the Software Engineering Body of Knowledge Version 3.0 (SWEBOK Guide V3.0)*. IEEE Computer Society Press.

Briand, Lionel, Yvan Labiche and L. O'Sullivan. 2003. "Impact Analysis and Change Management of UML Models." In *International Conference on Software Maintenance, 2003. ICSM 2003. Proceedings*. Amsterdam: IEEE, 256–65. doi: 10.1109/ICSM.2003.1235428

Broughton, Vanda. 2006. "The Need for a Faceted Classification as the Basis of All Methods of Information Retrieval." *Aslib Proceedings*: 58: 49–72.

Carlan, Eliana and Marisa Bräscher Basílio Medeiros. 2011. "Knowledge Organization Systems in the Information Science's approach." *Revista Ibero-Americana de Ciência da Informação* 4(2): 53–73.

Dal Forno, Gédson, Mário Borges and Felipe Martins Müller. 2017. "Relationship Between Critical Factors in Software Development Projects and the Attributes that Determine Their Success." *Revista Gestão & Tecnologia* 17: 172–90.

Dick, Jeremy, Elizabeth Hull and Ken Jackson. 2017. *Requirements Engineering*. 4th ed. New York, NY: Springer.

Faulk, Stuart R. 1997. "Software Requirements: A Tutorial." In *Software Requirements Engineering* 2nd ed., edited by R. Thayer and M. Dorfman,. IEEE Computer Society Press.

Fernandez, Daniel Mendez, Stefan Wagner, Marcos Kalinowski, Michael Felderer, Antonio Vetrò, Tayana Conte, Marie-Therese Christiansson et al. 2017. "Naming the Pain in Requirements Engineering: Contemporary Problems, Causes, and Effects in Practice." *Empirical Software Engineering* 22: 2298–338.

Garshol, Lars Marius. 2004. "Metadata? Thesauri? Taxonomies? Topic Maps! Making Sense of it All." *Journal of Information Science* 30: 378–91.

Gilchrist, Alan. 2003. "Thesauri, Taxonomies and Ontologies – an Etymological Note." *Journal of Documentation* 59: 7–18.

Gharib, Mohamad, Paolo Giorgini and John Mylopoulos. 2018. "Analysis of Information Quality Requirements in Business Processes, Revisited." *Requirements Engineering* 23: 227–49.

Goguen, Joseph A. 1996. "Formality and Informality in Requirements Engineering." In *IEEE International Conference on Requirements engineering Colorado Springs, Colorado.* IEEE Computer Society, 102–8. doi: 10.1109/ICRE.1996.10005

Gödert, Winfried, Jessica Hubrich and Matthias Nagelschmidt. 2014. *Semantic Knowledge Representation for Information Retrieval*. Berlin: De Gruyter Saur.

Grings, Luciana. 2015. "Authority Control at the National Library of Brazil: Brief History And Current Practices." *Revista Brasileira de Biblioteconomia e Documentação* 11: 139–54.

Hall, Tracy, Sarah Beecham and Austen Rainer. 2002. "Requirements Problems in Twelve Software Companies: An Empirical Analysis." *IEE Proc. Software* 149(5): 153–60.

Harpring, Patricia. 2010. *Introduction to Controlled Vocabularies: Terminology for Art, Architecture, and Other Cultural Works*. Los Angeles, CA: Getty Research Institute.

Hjørland, Birger. 2003. "Fundamentals of Knowledge Organization." *Knowledge Organization* 30: 87–111.

https://doi.org/10.5771/0943-7444-2022-6-411 - am 24.01.2026, 08:25:44. https://www.inlibra.com/de/agb - Open Access -

Hodge, Gail M. 2000. *Systems of Knowledge Organization for Digital Libraries: Beyond Traditional Authority Files*. Washington, DC: Digital Library Federation.

Jackson, Michael. 1995. "Problems and Requirements." In *Proceedings of 1995 IEEE International Symposium on Requirements Engineering (RE'95), York, UK*, 2–8. doi: 10.1109/ISRE.1995.512540.

Keßler, Carsten, Krzyzstof Janowicz and Mohamed Bishr. 2009. "An Agenda for the Next Generation Gazetteer: Geographic Information Contribution and Retrieval." In *17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '09 New York, NY, USA.* Association for Computing Machinery (ACM), 91–100. doi.org/10.1145/1653771.1653787

Kilov, Haim and Ira Sack. 2009. "Mechanisms for Communication between Business and IT Experts." *Computer Standards and Interfaces* 31: 98–109.

Kotonya, Gerald and Ian Sommerville. 1998. *Requirements Engineering: Processes and Techniques.* Chichester: Wiley

Kruchten, Philippe. 2003. *The Rational Unified Process: An Introduction*. Boston: Addison-Wesley Professional.

Lamsweerde, Axel Van. 2000. "Requirements Engineering in the Year 00: A Research Perspective." In *Proceedings of the 2000 International Conference on Software Engineering. ICSE 2000 the New Millennium, Limerick, Ireland*: IEEE, 5–19. doi:10.1145/337180.337184

Leffingwell, Dean and Don Widrig. 2003. *Managing Software Requirements*. Boston: Addison-Wesley Professional.

Lima, Gercina Ângela de and Benildes Coura Moreira dos Santos Maculan. 2017. "Comparative Study of Semantic Structures in Different Systems of Knowledge Organization." *Ciência da Informação* 46: 60–72.

Liu, Lin, Tong Li and Fei Peng. 2010. "Why Requirements Engineering Fails: A Survey Report from China." In *18th IEEE International Requirements Engineering Conference, Sydney, Australia*: IEEE. 317–22. doi:10.1109/RE.2010.45.

Lucio, Roberto Hernandez, Carlos Fernández Sampiere and María del Pilar Baptista Collado. 2013. *Métodos de Pesquisa*. Porto Alegre: Penso.

Mahmood, Khalid, Hironao Takahashi and Mazen Alobaidi. 2015. "A Semantic Approach for Traceability Link Recovery in Aerospace Requirements Management System." In *IEEE Twelfth International Symposium on Autonomous Decentralized Systems, Taichung, Taiwan*: IEEE, 217–22. doi:10.1109/ISADS.2015.33

McCulloch, Emma and George MacGregor. 2008. "Analysis of Equivalence Mapping for Terminology Services." *Journal of Information Science* 34: 70–92.

Morales-Ramirez, Itzel, Anna Perini and Renata S.S. Guizzardi. 2015. "An Ontology of Online User Feedback in Software Engineering." *Applied Ontology* 10: 297–330.

Nardi, Julio Cesar and Ricardo de Almeida Falbo. 2006. "Uma Ontologia de Requisitos de Software." In *9th Workshop Iberoamericano de Ingenieria de Requisitos y Ambientes de Software, La Plata, Argentina*. 111–24.

Nuseibeh, Bashar and Steve Easterbrook. 2000. "Requirements Engineering : A Roadmap." In *Proceedings of the Conference on The Future of Software Engineering, New York, NY.* Association for Computing Machinery (ACM), 35–46. https://doi.org/10.1145/336512.336523

Pressman, Roger. 2014. *Software Engineering: A Practitioner's Approach*. New York: McGraw-Hill Education.

Przybyłek, Adam. 2014. "A Business-Oriented Approach to Requirements Elicitation." In *9th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE), Lisbon*, edited by Joaquim Filipe and Leszek Maciaszek. IEEE, 1-12.

Ramalho, Vera Regina Casari Boccato Rogério Aparecido Sá and Mariângela Spotti Lopes Fujita. 2008. "The Contribution of Thesauri in the Construction of Ontologies as Tools for the Organization and Retrieval of Information in Digital Environments." *Ibersid: Revista de Sistemas de Información y Documentación* 2: 199–209.

Robertson, Suzanne and James Robertson. 2013. *Mastering the Requirements Process: Getting Requirements Right.* Upper Saddle River, NJ: Addison Wesley.

Rocha, Álvaro and José Braga de Vasconcelos. 2004. "A Framework to Analyse the Approach Adopted in the Information Systems Requirements engineering Activity." In *Proceedings of the International Conference on Software Engineering Research and Practice, SERP '04, Las Vegas, Nevada*, edited by Hamid R. Arabnia and Hassan Reza. CSREA Press, 573–9.

Ruy, Fabiano Borges, Ricardo de Almeida Falbo, Monalessa Perini Barcellos, Simone Dornelas Costa and Giancarlo Guizzardi. 2016. "SEON: A Software Engineering Ontology Network." In *20th International Conference on Knowledge Engineering and Knowledge Management (EKAW'16), Bologna, Italy*, edited by E Blomqvist, P. Ciancarini, F. Poggi and F. Vitali, Lecture Notes in Computer Science Vol. 10024. Cham: Springer, 527–42.

Sharma, Shilpa and Maya Ingle. 2011. "An Ontology Aided Requirement Engineering Framework." *International Journal of Advanced Computer Science and Applications* 2: 279–84.

Shiri, Ali and Keri Molberg. 2005. "Interfaces to Knowledge Organization Systems in Canadian Digital Library Collections." *Online Information Review* 29: 604–20.

Sitthithanasakul, Supavas and Noppon Choosri. 2016. "Using Ontology to Enhance Requirement Engineering in Agile Software Process." In *10th International Conference on Software, Knowledge, Information Management*

https://doi.org/10.5771/0943-7444-2022-6-411 - am 24.01.2026, 08:25:44. https://www.inlibra.com/de/agb - Open Access -

& Applications (SKIMA), Chengdu, China. IEEE, 181–6. doi:10.1109/SKIMA.2016.7916218

Solemon, Badariah, Shamsul Sahibuddin and Abdul Azim Abd Ghani. 2009. "Requirements Engineering Problems and Practices in Software Companies: An Industrial Survey." In *Advances in Software Engineering. ASEA 2009*, edited by D. Ślęzak, T. Kim, A. Kiumi, T. Jiang, J. Verner and S. Abrahão. Communications in Computer and Information Science Vol 59. Berlin; Heidelberg: Springer, 70–78.

Sommerville, Ian, and Pete Sawyer. 1997. *Requirement Engineering: A Good Practice Guide*. Chichester: Wiley.

Sommerville, Ian. 2016. *Software Engineering*. Harlow: Pearson.

Sowa John. 1991. *Principles of Semantic Networks: Explorations in the Representation of Knowledge*. San Mateo, CA: Morgan Kaufmann.

Stock, Wolfgang G. 2010. "Concepts and Semantic Relations in Information Science." *Journal of the American Society for Information Science and Technology* 61: 1951–69.

Sweis, Rateb. 2015. "An Investigation of Failure in Information Systems Projects: The Case of Jordan." *Journal of Management Research* 7: 173–85.

Tillett, Barbara B. 2004. "Authority Control: State of the Art and New Perspectives." *Cataloging and Classification Quarterly* 38(3/4): 23–41.

Tristão, Ana Maria Delazari, Gleisy Regina Bóries Fachin and Orestes Estevam Alarcon. 2004. "Faceted Classification System and Thesaurus: Technical Instruments for Knowledge Organization." *Ciência Da Informação* 33: 161–71.

Vital, Luciane Paula and Ligia Maria Arruda Café. 2011. "Ontology and Taxonomy: Differences." *Perspectivas Em Ciência Da Informação* 16: 115–30.

https://doi.org/10.5771/0943-7444-2022-6-411 - am 24.01.2026, 08:25:44. https://www.inlibra.com/de/agb - Open Access -