

RESEARCH ARTICLE

**THE USE OF ONTOLOGIES IN SOFTWARE  
ELICITATION: REVIEW**

**Abdalmoneim Mohamed Mohamed Khair<sup>1</sup> Farid Meziane<sup>2</sup>**

<sup>1</sup> Sudan University of Science & Technology, College of Graduate Study, Khartoum, Sudan

Email: [abdalmoneim122@gmail.com](mailto:abdalmoneim122@gmail.com)

<sup>2</sup> Data Science Research, Center ,University of Derby ,Derby DE221GB,UK

Email: [f.meziane@derby.ac.uk](mailto:f.meziane@derby.ac.uk)

HNSJ, 2021, 2(9); <https://doi.org/10.53796/hnsj2923>

**Published at 01/09/2021**

**Accepted at 25/08/2021**

**Abstract**

Research in recent years has probed the integration of research in the fields of Software Engineering and Semantic Web technologies in several domains. Semantic approaches in general have allowed a better understanding of domains that are known for their ambiguities and incompleteness. We have seen artificial intelligence in general being used to improve many stages of the software development process. Ontologies are being used successfully in the requirements engineering phase. This paper reviews some of the research related to the use of ontologies in requirements engineering (RE). Specifically review some approaches, methods, and tools. The main objective of this work is to show how ontologies are used in requirement elicitation which is then first step of (RE). Requirement elicitation is a crucial phase of (RE) and the use of ontologies to manage and facilitate the activities undertaken during this phase is of great importance. In summary, the main findings of this work are:

(1) there are empirical evidences of the benefits of using ontology's in (RE) in several domains for reducing ambiguity, inconsistency and incompleteness of requirements;(2) several research works have developed novel approaches for ontology usage in (RE); (3) some of which developed algorithms that improved and enhanced the (RE) process; (4) most studies addressed only functional requirements; (5) some studies focused on the use of ontologies in requirement modeling. Finally, we conclude this work by suggesting several areas of ontology in (RE) search opportunities in ontology and some issues that need to be addressed.

**Key Words:** *Requirements Engineering, Requirements Elicitation, Ontology, UML, Onto UML Natural Language Processing.*

## Introduction

The main objective of the software engineering discipline is to develop software and applications of high-quality satisfying users' requirements. The term Software Engineering (SE) was first coined in the mid-sixties and the two NATO sponsored conferences in 1968 was the official launch of the discipline and profession. Software Engineering has seen many developments and changes over the years at every step of the software development process. The simplest and probably the most used process for software development is the waterfall model [1]. The waterfall model, also referred to as the software life cycle model, describes the software process as a sequential set of phases where each phase depends on the outputs of the previous one. Sommerville describes the waterfall model as a plan-driven process [2]. Requirements Engineering (RE) which is the first phase of the waterfall model, is often cited as the phase where errors are expensive to correct and the source of many software failures. Many papers have been published to identify and describe the inherent problems of RE and the most recent include the process that analyzes the stakeholders' needs and identifies functional, non-functional and usability engineering requirements. In the last few years, another category of requirements namely security requirements, is added. user requirements are statements in a natural language plus diagrams of what services the system is expected to provide to system users and the constraints under which it must operate [2]. The RE phase is composed of any tasks and this comprises elicitation, analysis, specification, validation, and management of requirements. Requirements elicitation is the process of discovering, extracting, and gathering requirements for a system through communication with the stakeholders. Requirement elicitation from stakeholders and developers point of views is the crucial step in the requirements analysis process, and several methods, frameworks and tools have been developed to support human activities during this task, Goal oriented requirements analysis methods, scenario analysis, use case modelling techniques are examples of such approaches and methods. However, with all these developments, there are still many challenges in requirement elicitation. Hence, there is a need for more efforts and research to improve this phase and facilitate the dialog between the stakeholders and developers. In the context of software development, there is a growing interest in the use of ontologies.

The term ontology comes from the field of philosophy that deals with nature of beings [6] and introduced in computer science specifically artificial intelligence (AI) and defined as a set of representational primitives in a particular knowledge area [7]. Furthermore, Gruber [8] defined an ontology as an explicit specification of a conceptualization. It is within this understanding that ontologies are used in software engineering and specifically requirements engineering as their classes and properties are an abstraction of a specific domain [9]. Ontologies are becoming increasingly common to encode knowledge in many domains including software and requirements engineering. Ontologies are used in requirement engineering to resolve different types of problems for example in requirement elicitation, requirement validation, requirements disambiguation and requirements completeness. Ontologies are specified and developed using ontology languages such as the ontology web language (OWL).

The remaining of the paper is organized as follows. Section 2 we provide some background and definitions of the requirements elicitation phase. In section 3, we summarize the results of one of the most authoritative work in reviewing the use of ontologies in requirements engineering, that reviewed published works until 2013 and the paper itself is published in 2015. In section 4, we review some of more recent works on the use of ontologies in requirements engineering. The discussion is presented in section 4 and we draw the conclusions in section 5.

### **The requirements elicitation and analysis process**

In this paper, we are interested in the requirements elicitation and analysis task only. There are activities that precede this task such as feasibility study and other activities that follow this task such as requirements modelling and validation. Different stakeholders are involved in this phase and communication is sometimes challenging as they may have different priorities and may favor different notation and terminologies. Sommerville [2, pages 101- 102] identified four different activities in this task. The following subsections will provide a short description of these activities as defined by Sommerville [2].

### **Requirements discovery**

This phase is the start of the requirements engineering phase. Its purpose is to identify the system requirements from all the various requirements categories. The discovery

phase uses many techniques to elicit the requirements and include, the study of existing organizations' documents, the study of the requirements of the existing system. interview of the various stakeholders, questionnaires and in some instances observing how the current system and users operate and use the current system.

### **Requirements Classification and Organization**

It is known that during the requirements elicitation phase, stakeholders have different priorities for the future system requirements. Furthermore, some of these requirements are conflicting. This task is concerned with solving these two major issues. That of setting priorities for the elicited requirements and resolving the conflicts between the stakeholders. This resolution is conducted through negotiations and to agree on a compromise.

### **Requirements Specification**

In the requirement discovery activity, the requirements are developed in natural language (NL). During requirement specification, these requirements are transformed into a language that is acceptable by the developers and able to remove the ambiguities present in NL documents. Specifications can be developed using a graphical notation such as UML or in formal languages such as VDM and Z [10]. In the following section, we will review one of the most important early studies that reviewed works that use ontologies in requirements engineering.. The formatter will need to create these components, incorporating the applicable criteria that follow.

## **II BACKGROUND**

Dermeval et al. [11] conducted a review in 2015 on the applications of ontologies in requirements engineering. This is one of the strongest studies in this area and have reviewed publications up to 2013. They stated that some studies investigated the use of ontologies in software engineering and RE but did not achieve properly their goals. The aim of their study was to understand how ontologies were used in the requirements phase, which types of requirements they are supporting and whether ontologies have contributed to solving some of the problems that are inherent to requirements engineering. Their study aimed at answering seven research questions that are well defined and motivated. In their study they have used the categorization of the RE phases as defined by Kotonya and Sommerville [12] and these were (i) elicitation; (ii) analysis

and negotiation; (iii) specification; (iv) validation and (v) management . To the question which phases of the RE process are supported by ontologies, the study found that 83.6% of the studies supported Specifications, 58.2% supported Analysis and negotiations, 35.8% Management and 25.4% Elicitation and 6% validation. Note that some publications support more than one phase. To the question that identifies the types of software requirements modelling styles used together with ontologies the study found that 49.3% support textual requirements, 23.9% support UML, 19.4% scenario-based requirements, 17.9% support Goal- oriented requirements and the rest of the publications support other types of requirements' styles such as feature models, business process models and formal language. The question on the use of ontologies to describe functional and non-functional requirements, the study find that 52.24% support functional requirements, 2.99% for non-functional requirements and 44.78% support both. To the question of the contributions in ontology-driven RE i.e. the problems they are attempting to address, 56.72% are used for ambiguity, inconsistency and/or incompleteness detection; 35.82% for requirements management/evolution; 26.87% for domain knowledge representation and the other studies were used for other problems such as integration between requirements and architecture, requirements models interoperability and goal decomposition. For research question five, ontology-related languages used in RE, the majority of the ontologies identifies use the OWL language (49.3%), followed by SPARQL (10.4%) and description logics (7.5%). The rest of the ontologies use other less popular languages (RDF, XML, UML, XSPARQL etc). The purpose of the sixth research question was to identify the studies which reused ontologies developed in other studies in the context of RE. This is not of importance to this review. Finally, research question seven looked at the benefits of using ontology in the requirements process. 61.2% of the studies reported positive impact but without any empirical evaluation. 37.3% reported a positive impact with empirical evaluation. Only one study (1.5%) reported a negative impact with an empirical evaluation . To conclude, this study has shown that most efforts on the use of ontologies in the RE are for requirements specifications, mainly for functional requirements using the OWL language. Ontologies are mainly used with requirements written in natural language and used mainly to solve the ambiguity, inconsistency, incompleteness detection and

requirements management and evolution. In general, ontologies have a positive impact on RE but a large proportion of these studies were not empirically evaluated.

### III Related works

In this section, we will review some of the works that are published after 2013 and assess if there has been a shift in the use of ontologies in the requirements engineering process. Liu [13], developed a mechanism for automatically detecting conflicts in non-functional requirements. He developed a tool, CDNFRE, that include four essential elements namely metadata, ontology, cause-and-effect relationship, and conflict detection rule. The ontology was used to store concepts and semantic relationships. By applying ontology concepts, the system analyst can provide the metadata needed to model the non-functional requirements. This study was one of the few studies that attempted to deal with non-functional requirements and specifically dealing with conflicts during the maintenance phase. Nguyen et al. [14] proposed a framework that integrates goal and use cases. The GUI-F framework provides an automatic support for the detection and resolution of incompleteness, incorrectness and inconsistency in natural language goal and use case. The natural language specifications are transformed semi automatically into Manchester OWL Syntax for automated reasoning. The framework generates explanations for the detected problems and is able to suggest repairing alternatives. Provenzano et al. [15] proposed SARE, a new Safety Requirements Elicitation method that is based on a Hazard Ontology. The ontology is used to capture the knowledge about hazards identified during the safety analysis, and structured it as hazard's components and relations among them. This information is then used by the SARE approach to create a list of questions which guide the analyst in the requirements elicitation. Answering the questions is a part of the elicitation process to capture the safety requirements that are correct with respect to the hazard they are supposed to mitigate. Alkhamash [16], developed a prototype that generates requirements in the formal method By using an OWL ontology. His approach makes use of OWL ontologies to incrementally develop traceable Event-B models. Attempto Controlled English (ACE) is used to derive textual requirements from OWL ontologies and then map ACE-driven requirements to Event-B models. He applied his prototype in developing safe, secure, and reliable smart city systems. Alshehri et al. [17] proposed a

novel framework for the semantic processing of software requirements based on the concept of an ontology. The aim of their work is to enable software developers to elicit software requirements and implement structures that suit particular requirements. The proposed ontology has two components: requirement elicitation and reusable parts. The solution aims to incorporate an intermediate step involving conceptual design work. The additional step is actually a new method for mapping UML to OWL. This approach provides an opportunity for a formal representation of a domain knowledge. The challenge is that conventional engineering approaches do not rely on ontology representations using Semantic Web Languages. Therefore, the proposed solution will seek to adapt Protégé to support the addition of ontology languages. The solution involves two major steps. The first step is analyzing XSLT processors that is the .NET Framework Class Library and XML Spy to generate OWL files by using XSLT file as the input to enhance the mapping rules generated with Poseidon using OUP. H. Li, Ma and Yao [18] designed an ontology-based knowledge base to describe the credibility related concepts. The ontology-based knowledge is then used to elicit and model the trustworthiness requirements. Mahmoud et al I.[19] provided a framework for reducing requirements errors during earlier requirement engineering activities and achieving software project's success with high quality. Furthermore, the framework represented the software project requirements in double layer presentation. The first layer is the ontology layer that is constructed from textual requirements by using natural language processing software integrated with a java tool and protégé, an ontology modeling tool. The second layer is the concept maps layer; concept maps are generated. The first layer that contains ontology is a formal layer constructed based on a form of textual requirements to provide an explicit specification.

J.Yousef and Almarabeh [20] proposed a requirement elicitation framework starting with organization business organization process and build a CRUD matrix which provides all possible relationships between entities and functions of the system and generated relationship between entities and functions. This framework was evaluated using real case study the cancer care and registration in Jordan. Figure 2 illustrated the proposed framework. Elliott and Allen [21] Developed a methodology with automated support for producing software requirements specification that includes

requirements engineering data elements recommended in the Software Engineering Body of Knowledge (SWEBOK). Presented as seven use cases and an ontological framework, it also presents three empirical retrospective case studies that demonstrated the practicality of the methodology. The case studies also demonstrated that the ontology is readily customized for various application domains. They conclude that the ontological support is a promising way to enhance processes that produce a software requirements specification then build Ontology. The ontology also encoded data elements in the Software Requirements Knowledge Area of the SWEBOK. to extract software requirements from an OWL/RDF ontology and format them in simple text files. They then automatically create a software requirements specification document. Ermolayev [22] developed a methodology for evaluating the fitness of an existing ontology to the requirements of the knowledge stakeholders in a domain of requirement elicitation and demonstrated thorough indirect elicitation, ensuring completeness, correctness of interpretation, using in ontology evaluation of these requirements is a must for ontology engineering. The approach used in the reported research is based on the use of OntoElect – the methodology for ontology refinement and the workflow of ontology of OntoElect contains three phases: feature elicitation, requirements conceptualization, and ontology evaluation and elicits the set of terms extracted from a saturated collection of documents in the domain and using term significance in the form of numerical scores. Finally, the mappings are specified between the elements in the requirements and ontology elements .Nguyen et al. [23] developed a knowledge-based requirements engineering framework (KBRE) for eliciting, elaborate and detecting inconsistencies in requirements engineering. In KBRE domain knowledge and semantics of requirements are centralized. Nguyen et al [24] tool provided an automated analysis of goals and use cases for incompleteness, incorrectness and inconsistencies and also enables the generation of problem explanations and resolution alternatives. Their GUITAR tool is based on goal-use case integration meta-model and the ontology of the domain knowledge and semantics.de Graaf et al [25] described the usefulness of Software Architecture (SA) documentation that depends mainly on how well its Architectural Knowledge (AK) can be retrieved by the stakeholders in a software project. They have shown that the use of an ontology-based (SA) documentation,



describing different roles in software development have different needs for AK. This approach involves the use of typical questions for eliciting and constructing an ontology and depends on eight contextual factors especially in complex software projects with diverse AK users. A case study was used for acquiring and modeling AK needs. The results show this approach can improve the AK descriptions and retrieval. Mammari and Laleau [26] developed a goal-based approach in which the building of an initial formal model (in Event-B) is driven by a goal-oriented requirements engineering model (SysML/KAOS) and defined a set of rules to derive a partial Event-B to enhance the goal model in order to obtain a more complete formal specification and advocate the specification of a domain ontology to share common understanding of the structure of different applications to enhance and useful for complex systems to explicit and make clearer the domain knowledge use class and object diagrams are then specified to detail its components and their relationships and describe how the ontology and the structural model are translated into Event-B and illustrated through a landing gear system. Rashwan et al [27] developed a novel manually annotated (gold standard) corpus for sentence-based classification of requirements and focus on non-functional requirements. The corpus can be used for many automated analysis tasks and also worked improve semantic tool support for the domain of requirements engineering. Furthermore, they developed a new classification algorithm for the automatic categorization of requirements in software specifications . the results of this work will be of interest to researchers as well as practitioners from industry, who are interested in estimating the effort for building requirements in general and improving software quality in particular, and use measurement data in requirements engineering. In addition, they used the ontology to transform software requirements documents into a semantic representation automatically, and to estimate the cost of the software system and measure the quality of the written requirements. S. Morales-Ramirez et al. [28] developed a novel user feedback ontology specified in OntoUML and focused on online feedback given by users upon their experience in using a software service or application from different customer evaluation that is expressed in terms of quality–price comparison of similar products and then recommendation to people to buy (or not) the product as the summary of a product an ontology apply to online user feedback in

general added specific entities that are needed when we consider user feedback in software application domain. Rastgoole et al [29] developed an approach for automated generation of requirements ontology using UML diagrams in service-oriented architecture (SOA). Furthermore, the goal of this paper is to convenience progress of software engineering processes like software design, software reuse and service discovery. The proposed method is based on four conceptual layers. The first layer includes requirements achieved by the stakeholders, the second one designs service-oriented diagrams from the data in the first layer and extracts XMI codes from them. The third layer includes requirement ontology and protocol ontology to describe the behavior of services and relationships between them semantically. The fourth layer makes standard the concepts that exists in the ontology of the previous layer and also this method can extend to other design diagrams such as sequence diagrams and activity diagrams. Sanya and Shehab [30] developed a novel knowledge-based engineering (KBE) framework that enabled product design systems within the aerospace industry and the aim of this paper is to strengthen the structure, reuse and portability of knowledge consumed within (KBE) the system. The proposed framework uses an ontology-based approach for semantic knowledge management and adopts a model-driven architecture and uses the following phases (1) Capture knowledge required for KBE system; (2) Ontology model construct of KBE system; (3) Platform-independent model (PIM) technology selection and implementation and (4) Integration of PIM KBE knowledge with computer-aided design system .and also this paper has illustrated the need to address the portability quality attribute/non-functional requirement of a KBE system and ensure this is reflected in the KBE system architecture.

## Discussions

This section starts with a discussion about the scope of this review and summarize some of related works that uses or developed approaches or method used in requirement engineering (RE) using ontology. The scope of the review is focuses on how ontology's are employed in RE, in terms of their use in the RE process of software requirements modeling, ontology languages, framework and usage in requirement elicitation. All of the papers in this review uses the ontology to develop methods or to build algorithms to enhance the requirement engineering. The use of the ontologies has shown to solve

many of the classical issues inherent to requirements engineering such as incompleteness, ambiguities, and validation. The ontologies have expanded to use of knowledge engineering that is used in the eighties and nineties, as an approach to improve software requirements that remains albeit many efforts the most problematic phase of software engineering. The introduction of AI in software engineering in general and requirements engineering in particular has improved most of the phases of the software engineering process [30]. The development of domain specific ontologies to support requirements elicitation is a research direction that can provide further improvements and solid solution to current problems.

### **Conclusions**

The main goal of the paper is to improve the understanding of how ontologies support requirement engineering to identify evidence of its use in this field of requirement engineering. About ten papers presented a novel methods and approaches used in requirement engineering using ontology to enhance the requirement engineering elicitation and avoid ambiguity in terms and inconsistency, incompleteness of requirement engineering. Some of the papers also show the framework use in requirement engineering to develop the process of collection of requirements and modelling using tools like UML (Unified modelling language) and also some of the papers used OWL (ontology web language) to write the requirements engineering to help the stakeholders and developers to understand the requirements engineering. There seems to be more interest in the last few years to work on requirements elicitation issues instead of requirements specifications. Another aspect is the interest in non-functional requirements and further developments of domain specific ontologies. The use of OWL as the language for the description and specification of ontologies remains the language of choice.

### **References**

- [1] R. Winston (1970), "Managing the Development of Large Software Systems", Proceedings of IEEE WESCON, 26 (August): p. 1–9.
- [2] I. Sommerville (2011), "Software Engineering (9th Edition)". Pearson.
- [3] T. Hall, S. Beecham, A. Rainer (2002), "Requirements Problems in Twelve Software Companies: An Empirical Analysis". IEEE Proceedings of Software 149(5), 153–160.

- [4] B. Solemon, S. Sahibuddin and A.A.A Ghani (2009), “Requirements Engineering Problems and Practices in Software Companies: An Industrial Survey”, In: Ślęzak D., Kim T., Kiumi A., Jiang T., Verner J., Abrahão S. (eds) *Advances in Software Engineering (ASEA 2009)*. Communications in Computer and Information Science, vol 59. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-10619-4\\_9](https://doi.org/10.1007/978-3-642-10619-4_9).
- [5] J. Iqbal, R.B. Ahmad, M. Khan, F. Amin, S. Alyahya, M. H. N. Nasir, A. Akhunzada, M. Shoaib (2020), “Requirements engineering issues causing software development outsourcing failure”. *PLoS ONE* 15(4): e0229785. <https://doi.org/10.1371/journal.pone.0229785,2020>
- [6] X. Chen, Z. Jin, L. Yi (2007), “An Ontology of Problem Frames for Guiding Problem Frame Specification”, In: Zhang Z., Siekmann J. (eds) *Knowledge Science, Engineering and Management. KSEM 2007*. Lecture Notes in Computer Science, vol 4798. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-540-76719-0\\_38](https://doi.org/10.1007/978-3-540-76719-0_38)
- [7] P. Mika, H. Akkermans (2004), “Towards a new synthesis of ontology technology and knowledge management”, *Knowledge Engineering Review*, 19:317–345
- [8] T.R. Gruber T.R (1993), “A translation approach to portable ontology specifications”, *Knowledge Acquisition*, 5(2):199–220
- [9] N. Guarino (1998), “Formal ontology in information systems”, *Proceedings of FOIS'98*, Trento, Italy, 6-8 June, IOS Press, Amsterdam.
- [10] F. Meziane, N. Athanasakis and S. Ananiadou (2008), “Generating Natural Language Specifications from UML Class Diagrams”,
- [11] *Requirements Engineering Journal*, 13(1):1-18. Diego Dermeval, Jéssyka Vilela, Ig Ibert Bittencourt, Jaelson Castro, Seiji Isotani, Patrick Brito & Alan Silva (2016). “Applications of ontologies in requirements engineering: a systematic review of the literature”. *Requirements Engineering*, 21, 405–437. <https://doi.org/10.1007/s00766-015-0222-6>
- [12] G. Kotonya, I. Sommerville I (1998) “Requirements engineering—processes and techniques”. Wiley, New York

- [13] Chi-Lun Liu (2016), "CDNFRE: Conflict detector in non-functional requirement evolution based on ontologies". *Computer Standards & Interfaces*, 47, 62-76, ISSN 0920-5489.
- [14] T.H. Nguyen, J.C. Grundy and M. Almorisy (2016), Ontology-based automated support for goal–use case model analysis. *Software Quality Journal*, volume 24, 635–673.
- [15] L. Provenzano, K. Hänninen, J. Zhou and K. Lundqvist (2017), "An Ontological Approach to Elicit Safety Requirements," 2017 24th Asia- Pacific Software Engineering Conference (APSEC), Nanjing, China, 2017, pp. 713-718, doi: 10.1109/APSEC.2017.91.
- [16] E. Alkhamash (2020). "Formal modelling of OWL ontologies-based requirements for the development of safe and secure smart city systems". *Soft Computing* 24, 11095–11108. <https://doi.org/10.1007/s00500-020-04688-z>
- [17] A. Alshehri, M. Basher, R. Qureshi, "Proposed Framework to Manage Software Requirements and Reuse", *International Journal of Modern Education and Computer Science (IJMECS)*, 9(12): 49-56, 2017.DOI:10.5815/ijmeecs.2017.12.06.
- [18] R. Li, S. Ma and W. Yao, "Ontology-Based Requirements Generation for Credibility Validation of Safety-Critical System," 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, Liverpool, UK, 2015, pp. 849-854, doi: 10.1109/CIT/IUCC/DASC/PICOM.2015.126.
- [19] [19 Mahmoud Abd Ellatif, Marwa Salah Farhan, Mohamed Kamel Elmahy, Karama Ali Mohamed,"Using Ontology-Based Concept Maps to Reduce Requirement Engineering", *International Journal of Computer Science and Information Security (IJCSIS)*, Vol. 15, No. 6, June 2017.
- [20] Rana Yousef and Tarama Almarabeh (2015), "An enhanced requirement elicitation framework based on business process model", *Scientific Research and Essays*, 10(7):279-286.
- [21] Robert A. Elliott Sr and Edward B. Allen (2016), "Creating a Software Requirements Specification Document Using an Ontology Based Methodology",

International Journal of Advanced Research in Science, Engineering and Technology 3(9): 2616-2630.

- [22] Vadim Ermolayev (2018), "OntoElecting Requirements for Domain Ontologies", International Journal of Conceptual Modeling, vol. 13 pp. 86-109. DOI:10.18417/emisa.si.hcm.9.
- [23] Tuong Huan Nguyen, Bao Quoc Vo, Markus Lumpe, John Grundy (2014), "A framework for knowledge-based requirements engineering.", Software Quality Journal 22:87–119
- [24] Tuong Huan Nguyen, John Grundy, Mohamed Almorsy (2014), "GITAR: An Ontology-based Automated Requirements Analysis Tool", Karlskrona, Sweden, IEEE, 2014
- [25] K.A. de Graaf, P. Liang, A. Tang, W.R. van Hage, H. van Vliet (2014), "An exploratory study on ontology engineering for software architecture documentation", Computers in Industry, 65(7): 1053-1064, ISSN 0166-3615, <https://doi.org/10.1016/j.compind.2014.04.006>.
- [26] Mammar A., Laleau R. (2016) On the Use of Domain and System Knowledge Modeling in Goal-Based Event-B Specifications. In: Margaria T., Steffen B. (eds) Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques. ISoLA 2016. Lecture Notes in Computer Science, vol 9952. Springer, Cham. [https://doi.org/10.1007/978-3-319-47166-2\\_23](https://doi.org/10.1007/978-3-319-47166-2_23).
- [27] A. Rashwan, O. Ormandjieva and R. Witte (2013), "Ontology-Based Classification of Non-functional Requirements in Software Specifications: A New Corpus and SVM-Based Classifier," 2013 IEEE 37th Annual Computer Software and Applications Conference, Kyoto, Japan, 2013, pp. 381-386, doi: 10.1109/COMPSAC.2013.64.
- [28] Abderahman Rashwan , Olga Ormandjieva , Rene Witte (2013), "Ontology-Based Classification of Non-Functional Requirements in Software Specifications: A new Corpus and SVM-Based Classifier", IEEE 37th Annual Computer Software and Applications Conference.
- [29] Vahid Rastgoo, Monireh-Sadat Hosseini, Esmail Kheirkhah (2014), "Semantic web-base software engineering automated requirements ontology generation in

SOA", International Journal of Web & Semantic Technology (IJWesT) Vol.5, No.2, 2014.

- [30] I.O. Sanya ,E.M. Shehab (2014), "An ontology framework for developing platform-independent knowledge-based engineering systems in the aerospace industry", International Journal of Production Research, Vol. 52, No. 20, 6192–6215,2014.
- [31] F. Meziane and S. Vadera (Eds) (2009), Artificial Intelligence Applications for Improved Software Engineering Development: New Prospects, IGI publishing. ISBN: 978-1-60566-758-4.