A SYSTEMATIC LITERATURE REVIEW OF THE ROLE OF ONTOLOGY IN MODELING KNOWLEDGE IN SOFTWARE DEVELOPMENT PROCESSES

By Evi Triandini

IPTEK The Journal of Technology and Science, 32(3), 2021 (e/pISSN:2088-2033, 0853-4098)
DOI: 10.12962/j20882033.v32i3.12998 | Received 9 Sept, 2021; Accepted 12 Dec, 2021

REVIEW PAPER

A SYSTEMATIC LITERATURE REVIEW OF THE ROLE OF ONTOLOGY IN MODELING KNOWLEDGE IN SOFTWARE DEVELOPMENT PROCESSES

Evi Triandini*[1] | Marco Ariano Kristyanto[2] | Ravi Vendra Rishika[3] | Franky Rawung[4]

[1]Dept. of Information System, Institut Teknologi dan Bisnis STIKOM Bali, Denpasar, Indonesia

[2]Information Technology Helpdesk, Universitas Ciputra Surabaya, Surabaya, Indonesia

[3]Anti Fraud Platform Engineering, Information Security Division, Bank Rakyat Indonesia, Jakarta, Indonesia

[4]Dept. of Training, Creative Media, Surabaya, Indonesia

Correspondence

*Evi Triandini, Dept of Information System, Institut Teknologi dan Bisnis STIKOM Bali, Denpasar, Indonesia. Email: evi@stikom-bali.ac.id

Present Address

Jl. Raya Puputan No.86, Dangin Puri Klod, Kec. Denpasar Timur, Denpasar 80234, Indonesia

Abstract

Ontology in software development is explained as presenting the properties of things within a domain knowledge and how they are interrelated to each other by defining a set of notions and taxonomies that exemplify the knowledge. It is used to determine the ambiguity in the software requirements specification. Though averred to be useful, the software engineering communities are still unfamiliar with the role of Ontology in modeling knowledge in software development processes. Moreover, not much has been known about the role of Ontology in software engineering processes. The objective is to map and explain the substantiation about the role of Ontology in Modelling Knowledge and the challenge faced by the software engineering team to understand how far ontology can help them determine the ambiguity in modeling and software development processes. We have carried out a methodical review of the literature issued between 2012 and 2021 and recognized 150 publications that talk over the role of ontology in modeling knowledge in software development processes. This study conveyed and employed particular inclusion and exclusion criteria in bi-rounds to establish the utmost pertinent publications for our research objective. The review acknowledged 22 applications that explain ontologies' primary role in software development processes. However, our findings suggest ontology's role in software engineering as a investigation background requires extra consideration. A further experimental result I needed to better understand the role of ontology in modeling knowledge in software development with quality requirements as well as self-organizing groups.

KEYWORDS:

Knowledge Modeling, Model Ambiguity, Ontology, Software Development Process, Systematic Literature Review

1 | INTRODUCTION

In computer science, the phrase ontology refers to a metadata model representing a set of notions or conceptions within a specific subject area and the relationship between those conceptions. In software engineering, ontology is used to do many things, for example, domain analysis [1]. Another function of ontology is to detect the ambiguities in software engineering [2]. During the software development cycle, the role of ontology has been improved in modeling knowledge. As we mentioned, ontology plays an essential role in detecting ambiguities in software engineering. Ontological representation is divided into two stages: at the start translating the database rules corresponding to ontology syntax and schema and finding a generalization of the ontology model [3]. The need for complex systems and software always grows. However, the risk involved in the building a software also rises. The risk of system failure, unsuitable requirements, and faults as occurred in software engineering processes. The issue is that every reference model employs its vocabulary to deal with this phenomenon [4].

This paper conducts a methodical literature assessment to help researchers solve the issue. The article's structure is as follows. The next part previews the previous literature review on the functionalities of ontology in software engineering processes and identifies the gap in the literature and a need for a deeper investigation. The third part explains the research questions and the systematic approaches followed for the current services in the role of ontology in software modeling. The fourth part summarizes the main finding of our research. The fifth part presents the discussion of the result. The last part deduces the result of the studies, describes consequences for scholars and real-world practitioners, as well as outlines the study's limitations.

2 | PREVIOUS RESEARCHES

In the research areas of software development, there have been limited number of works related on investigating the ontology methods (summarized in Table 1), such as ontology usability issues (Isotani et al. [5]), software testing ontology (Tebes et al. [6], and ontology representation of UML (Mkhinini et al. [7], Verdonck et al. [8]).

Isotani et al. [5] conducted a methodical literature assessment on the use of ontologies and techniques benefit in the Software Engineering area. The study focused on usability issues in the ontology model concerning architecture. Their discovery indicates usability issues in the ontology model for research interdisciplinary in the area of Ontology and Software Engineering, as a result, generate better development methods or tools and provide more robust, better quality, and satisfying software for the requirements of customers and users.

Tebes et al. [6] performed a methodical literature assessment to explain software testing ontologies. It revealed that the software testing ontology minimizes the existing heterogeneity, indistinctness, and imperfectness problem in attributes and relations. Tebes could not find any of direct link between the conceptualized ontologies and the Non-Functional Requirements (NFRs) and Functional Requirements (FRs) statements.

Mkhinini et al. [7] carried out a methodical literature assessment on combining UML and ontology. The study investigates the underlying paradigm, and the approaches to merging both modeling concepts are becoming progressively emergent. The paper aims to arrange for a broad summary of the two domains by carrying out a methodical literature assessment of the pertinent research studies. They also created a survey to find the inter-association amongst UML and ontology based on the hypothetical and applied viewpoints. They show a thorough grouping of the existing studies based on well-thought-out concerns and hand-on cases.

Verdonck et al. [8] examined Ontology-Driven Conceptual Modeling (ODCM) in software engineering. The literature review contributed to analyzing the distinctive novelty of the studies and ascertaining their current advancement by arranging, assessing, and deducing pertinent studies to time correlated to ODCM.

3 | RESEARCH METHODOLOGIES

This study adopts the strategies recommended by Kitchenham et al. [9] in the research process. Consequently, this study presents the primary step in our methodical assessment: plan, implement, and report the assessment results.

TABLE 1 Summary of selected literature on ontology model in software development reviews at large.

Study	Goal	Research Question/Goals
Isotani et al.[5]	Combining ontology methods with semantic techniques	Develop methods, techniques, and environments to facilitate the production of semantic software using a interdisciplinary approach.
Verdonck et al.[8]	A review of Ontology-driven conceptual modeling:	Do the practitioner publish in the IEEE ontology conference how the ODCM impacts software engineering
Tebes et al.[6]	Explain software testing ontologies	How are the software testing ontologies changed overtime
Mkhinini et al.[7]	Combining UML and Ontologies	What usability issues are used in combining UML and ontologies?

TABLE 2 The search sources.

Type	Source
Electronic Source	Elsevier Google Scholar Mendeley IEEE explorer
Searched item	Journal, conference paper, workshop
Search Applied on	Full text - to avoid missing any of the papers that do not include our research keyword in titles or abstracts but are relevant to the review object
Language	English
Publication Period	From 2012 to 2021

3.1 | Planning the review

The design of this assessment study was based on a set of proposed research questions pertinent to the predefined study goals. This study also determines the search strategies, keyword-search strings, and insertion and omission criteria. The following elaborates the process in step-by-step manner.

3.1.1 | Review objectives and research questions

With the emerging application of ontology in software engineering, we considered that exploring and reviewing the vital roles of ontology in software engineering. . Ontological representation is divided into two stages: at the start translating the database rules corresponding to ontology syntax and schema and finding a generalization of the ontology model [3]. Thus, the primary objective of our study is to understand the roles of ontology in modeling knowledge in software development and the challenge that teams face when analyzing the role of ontology in software development. We also thought that ontology modeling practices could resolve the challenge in software engineering and the software development process.

In order to achieve our goals, we formulated the research questions. We designed the subsequent research questions:

RQ-1: What is the role of ontology in software modeling?

RQ-1: What are the challenges of Ontology in software engineering?

3.1.2 | Search Strategy

The literature review by Inayat et al.[10] was used as the main guideline for this research. After reviewing and determining the research questions and goals. We began by formulating two formal search strategies to capture all possible firsthand-specific publications for the objectives of our research. The detailed design in this research on determining the search area can be seen in Table 2. The search area comprised e-databases, e-journals, and e-proceedings, Based on the initial result, we expanded the coverage by finding the related publications through citations and references. This strategy is called Snowballing (Forward and Backward) method. It identifies other meaningful and comprehensive studies. This approach strategy aims to add as much potential research works as possible. Therefore, our analysis might be deeper and more actual on the topic we have set in this research. Afterward, we employ filtering by applying a set of criteria that determine whether a study should be included or excluded from the list. This filtering would be carried out as separate rounds and involves several scholars, as explained in Section 3.1.4.

3.1.3 | Search Criteria

In this study, we grouped the search criteria into two separate groups. Each group consists different set of keywords to be used for search the publications. The grouping is as follows.

1. C1
   C1 is the string of keywords related to the Role of Ontology in Modeling Knowledge in Software Development Processes, such as Ontology, Ontology Role, Software Modeling, and Software Development.

2. C2
   C2 is the string of keywords related to Software Development Process or Software Engineering, such as Engineering, Software Engineering.

We entered each keyword (search string) separately into each electronic database. The basis for the manual process is the search functionality provided by each electronic database. The search process was considered as a learning and experimenting process for our methodical literature assessment study.

3.1.4 | Inclusion and Exclusion Criteria

To achieve a common agreement whether a publication should be considered or omitted from the list, we follow a set of criteria. The criteria can be classified as two separated groups as follows.

- **Inclusion criteria:** (I1) It should want through a pee-reviewing process; (I2) It should be written in English; (I3) It should be pertinent to the keyword terms defined in search criteria; (I4) It should be an original paper, an experimental or firsthand report, or part of a proceedings; and lastly (I5) It should be issued between 01-2012 and 07-2021.
- **Exclusion criteria:** (E1) Any publication which does not emphasis on exactly on ontology methods but merely discuss the ontology software engineering as a secondary issue (e.g., it cites ontology as a modifier of a term mentioned in the publication); (E2) Any publication that does not talk over the vital part played by ontology in software engineering; (E3) Any publication that does not abide to the inclusion criteria; and lastly (E4) Subjective view, believes, assessment, invited speech, comparison study, editorial notes, comment from editor or reader, tutorials, forewords, lecture notes, and non-scientific publications.

3.2 | Conducting the Review

In this part, we present our findings from the publication extraction process and filtering process of the extracted publications.

3.2.1 | Study Search and Selection

Following the search strategy mentioned in section 3.1.2, the selected electronic databases were searched and the studies retrieved. In this original search, we retrieved 150 studies, as shown in Table 3. It is important to note that we only selected databases that publish peer-reviewed papers. An extensive inspection of the study titles and abstracts was made by one of the researchers (Round-1) by applying inclusion criteria. Most retrieved studies fell within the inclusion criteria (I2, I3, and I5). Due to the limitation of search engines in using the search string to the entire body of the text of the paper, a substantial number of results retrieved were then discarded. As a result of this first-round classification, we ended up with 72 candidate studies. We also ensured that the papers were not lectures notes, other SLR articles, discussions, tutorials, preface, and presentation. Then in Round-2, the pre-selected papers were assessed by the second (one of the co-authors) and third (independent and experienced researchers) to apply the exclusion criteria (E1, E2, E3, E4). We conducted an online scoring and consensus meeting to review the agreements and disagreements raised between the researchers. The three researchers read the abstract and excluded the studies based on the exclusion criteria for the papers where consensus was not reached. Out of the 72 studies pre-selected after applying the inclusion criteria, 17 were excluded on the ground, and they do not discuss any topics directly related to the scope

TABLE 3 The number of identified studies during the distinct rounds of our systematic search.

#	Database	Retrieved	Round-1		Round-1	
			Included	Excluded	Included	Excluded
1	Elsevier	114	50	64	25	25
2	Google Scholar	4	3	1	3	0
3	IEEE	25	17	8	16	1
4	Mendeley	3	0	3	0	0
5	IEICE	2	1	1	1	0
6	Springer Link	2	1	1	0	1
Total		**150**	**72**	**78**	**45**	**27**

of our investigation (E1 to E4). They all referred to discussing the role of ontology outside of software engineering. Therefore our final selection consists of 45 studies (see the two rightmost columns in Table 3). The complete studies will be available in Appendix A at the end of the paper

3.2.2 | Data Extraction and Synthesis

According to the guidelines written by Inayat et al. [10], we defined a data extraction process to identify relevant information from the 45 included primary studies that pertain to our research questions. Our data extraction process begin by setting up a form to record ideas, concepts, contributions, and findings of each of the 45 studies. Using this form ensures subsequent higher-order interpretation. The following data were extracted from each publication: (i) review date; (ii) title; (iii) authors; (iv) reference; (v) database; (vi) relevance to the topic, i.e., the role of ontology, modeling knowledge, software development process; (vii) methodology (interview, case study, report, survey); (viii) data analysis; (ix) validation Techniques; (x) future work; (xi) limitations; (xii) country or location of the analysis; (xiii) year of publication.

We carried out the impartial quality assessments for the 45 publications, This process followed the guidelines described in part 3.1.4. Any disputes or differences among the assessors would be resolved over an informal consultation or dialogue.

3.2.3 | Methodological Quality Assessment

These methodical literature assessment use the publication quality attributes based on Inayat et al. [10] to assess methodological quality researches carefully chosen for the assessment. We used these quality assessment attributes to evaluate the value of each selected publication discoursing the vital part of ontology in software engineering activities. We represented the quality attributes as a set of inquiries that quantify the degree to which a publication is acceptable and may expand the extent of the research. Those inquiries should include at least three important attributes of good publications, i.e. thorough, trustworthy, and significant. Theses attributes have been chosen in order to enable abstraction and synthetizing the primary studies based on its expediency and the interpreting them Kitchenham et al. [9]. Several recent systematic reviews previously employed the quality measure associated with these criteria.

We evaluated each study based on the enquiries shown in Table 4 for determining categories and ranks. To ensure that each scholar would categorize and rate each publication with ease and consistently, we utilized a specified order or rank in the following series: 0, 0.5, and 1.0. This scale can be seen in Table 4. For the first inquiry (C1), the assessor would have to assess the quality of research aim of a publication. Is it explicitly stated in the paper? Is it clearly defined? Overall, the aims of around two-third of the publications are considered clear and well-defined. For the second inquiry (C2), the assessor would have to assess the quality of research background presentation of the publication. Is it explicitly stated in the paper? Is it well addressed? Overall, the research background presentation of two-third the publications are considered well-addressed. For the third inquiry (C3), the assessor would have to assess the quality of the finding presentation of the publication. Is it explicitly stated in the paper? Is it clearly presented? Overall, the finding presentation of less than half of the publications are considered clearly presented. For the fourth inquiry (C4), the assessors would have to assess the contribution degree of the publication. How significant is the publication would contribute to science? How significant is the practical implication of the work to the real world? The normalized score of selected studies which are based on quality scores, are shown in Figure 1 .

TABLE 4 Quality Assessment of the selected studies.

Quality Criteria from Obtained Studies	Categories and Scales	Grades
C1: Does the publication clearly define its research aim?	(1.0,yes; 0.5,hesitant; 0.0,no)	30 studies, 66.67%
C2: Does the publication present the research background accordingly?	(1.0,yes; 0.5,hesitant; 0.0,no)	30 studies , 66.67%
C3: Does the publication present the finding clearly?	(1.0,yes; 0.5,hesitant; 0.0,no)	21 studies , 46,67%
C4: Based on the findings, what is the contribution level off the research?	>80% =1, <20% =0, in between = 0.5	

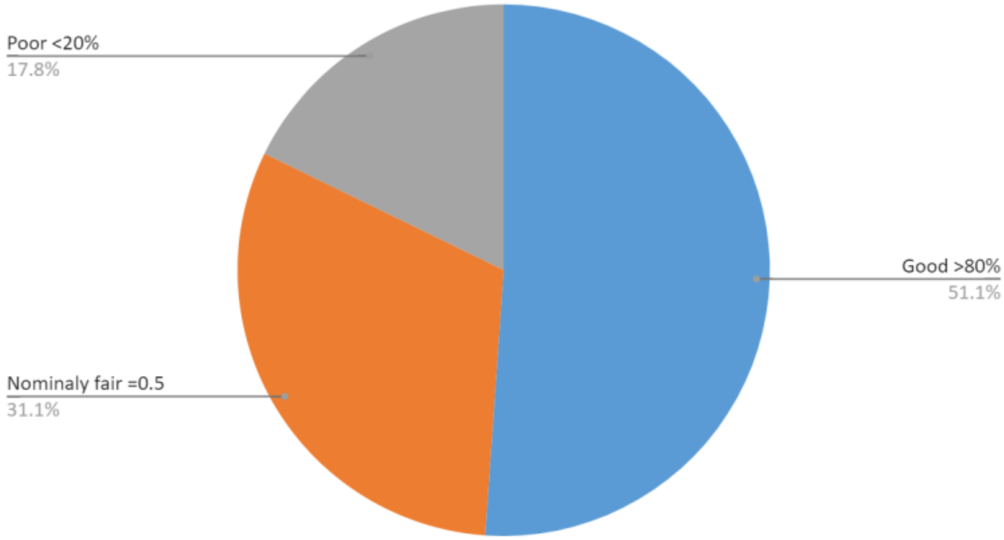

FIGURE 1 The distribution of the studies group by their quality scores (in percentage).

From Figure 1, it can be seen that 51.1% of total papers passed c4 with a score of 1. It means that research is valuable in future works, and it can be continuous by other researchers. There is 17.8% of total papers that do not pass the criteria in C4 because of duplicate research, and the remaining 31.1% of papers are nominally fair.

4 | FINDINGS OF OUR REVIEW

This section describes the overview of the studies and the findings of our assessments in light of each research question defined in the review plan.

4.1 | Overview of Studies

As we have stated in the section 3.2.3., this study has recognized 41 researchers. Table 5 suggests the disbursement of electronic databases which published the selected studies. All forty-one types of research aforementioned have been posted in a total of 30 publication venues. Ten of these venues are worldwide and us of a-unique conferences, and the remainder of venues are global and us of a-unique journals. As a bring about Table 5 additionally shows that the research is similarly distributed throughout the publication venues. In truth, each book source has only single or more articles issued within the publication. This indicates that there should be no solitary source preferred on ontology topics by software engineering authors.

Concerning the years of publications, we did not discover any massive research associated with our subjects before 2012. The distribution of the supplied paper at the side of the topics of our investigations. In our set of 41 studies, we found that the

TABLE 5 The publication distribution of the studies.

Distinctive Publisher	Type	Cited	Avg-Cited	Num-Papers
Procedia Computer Science	Journal	11	1.57	7
IEEE Access	Journal	15	3.75	4
Data & Knowledge Engineering	Journal	1	0.50	2
Journal of Information Security and Applications	Journal	1	1.00	1
Expert Systems with Applications	Journal	40	40.00	1
IEICE Transactions on Information and Systems E103.D	Journal	0	0.00	1
Advanced Engineering Informatics	Journal	38	38.00	1
IEEE Transactions on Systems, Man, and Cybernetics	Journal	16	16.00	1
Information and Software Technology	Journal	2	2.00	1
Computers in Industry	Journal	19	19.00	1
SoftwareX	Journal	0	0.00	1
School of Computing and Engineering	Journal	3	3.00	1
Computer Science	Journal	2	2.00	1
Journal of Systems Architecture	Journal	42	42.00	1
Science of Computer Programming	Journal	14	14.00	1
Computer Communications	Journal	12	12.00	1
Dyna	Journal	7	7.00	1
Information Systems	Journal	33	33.00	1
Future Generation Computer Systems	Journal	8	8.00	1
Computer Standards & Interfaces	Journal	18	18.00	1
International Conference on Computing for Sustainable Global Development	Conference	0	0.00	2
Conference of Open Innovations Association	Conference	1	1.00	1
IEEE International Conference on Computer Sciences and Information Technologies	Conference	1	1.00	1
IEEE International Conference on e-Business Engineering	Conference	1	1.00	1
International Conference on Computing for Sustainable Global Development	Conference	0	0.00	1
IEEE Canadian Conference on Electrical and Computer Engineering	Conference	3	3.00	1
International Conference on Software Paradigm Trends	Conference	31	31.00	1
International Conference on Information Technology and Nanotechnology	Conference	1	1.00	1
International Conference on Knowledge Engineering and Applications	Conference	0	0.00	1
International Conference on Computer, Information and Telecommunication Systems	Conference	0	0.00	1
Total				**41**

majority of our studies were affiliated with certain nations within an unmarried to take a look at. But, it is not tangible to decide the authorship's geographical location according taking a look. We also plotted the geographical distribution of authors inside the studies. Most of the studies come from South Asian nations, consisting of India and Pakistan, and the maximum range from Brazil to South America.

Aside from the overall studies published in every publication venue, we additionally offer the entire quotation and mean of citation of every research. For example, Table 5 shows 7 of forty-one papers are published in Procedia Computer Science, with the entire citation being 11 citations. This indicates Procedia Computer Science has an average citation of 1.57, calculated of general quotation divided through overall Papers published in Procedia Computer Science. From the 41 papers that we have amassed, the most citations are 42 citations for a paper published in the Journal of Systems Architecture. The average citation reached as high as 42.00 because it's far the best paper we amassed from this publication source.

Figure 2 shows the trend and growth of ontologies in software engineering from 2012 to 2021. From that graph, it can be seen that from the year 2016 to 2019, there may be an increasing trend in the studies about ontology in software engineering. Therefore, we can conclude that many researchers possibly have implemented the role of ontology in software research or software development discipline. This trend may continue within the next years due to the researchers and practitioners having understood how critical the role of ontology is in the software engineering discipline. The red-colored line in Figure 2 indicates a tendency for this topic to increase between 2019 to 2021 and within the following years.

Figure 3 shows the type of distribution of selected studies. We divided our collected papers into three categories of research. The first is Software Research, which approaches those papers containing the new proposed thoughts of using ontologies in software engineering to solve the hassle. The proportion of this topic is 14.6% of the total papers. The second is Software Development. It is the most dominant subject matter of the paper we have collected. It has a total percentage of 56.1% of our overall papers. And the closing is Software Practices, which amassed 29.3% of the entire papers. This means that 23 of 41 papers explained the role of ontology in Software Development and indicated that ontology has been applied and has a significant role in Software Development.

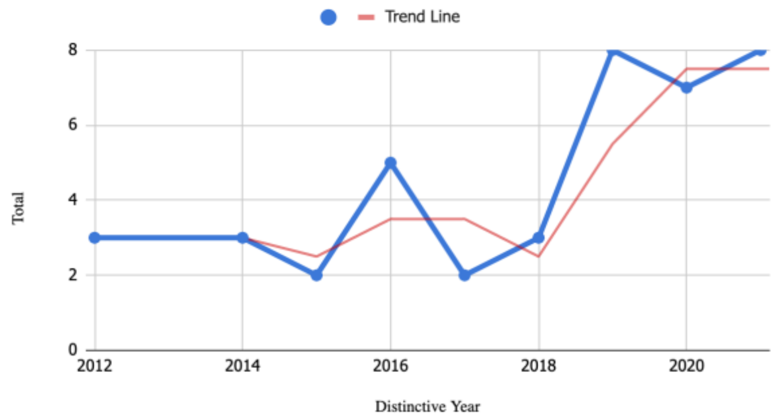

FIGURE 2 The study distribution based on its publication year.

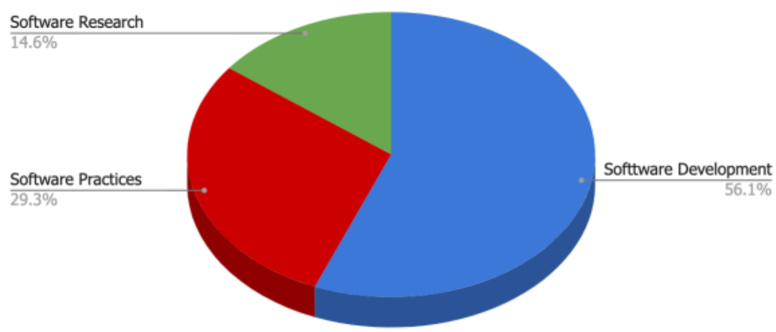

FIGURE 3 The study distribution based on their topics.

Figure 4 shows the map of the studies based on the country. The green-colored country significantly contributes to the studies, and the red-colored country suggests a low contribution. This graph is also associated with Figure 2 . It suggests the productive researchers from several countries who have published their work related to our subject. For numerous next years, there can be a variety of research about this topic internationally.The scale of this map is from 1 to 5. South Asia nations have ruled the studies in general of 41 papers that we have accumulated, with India and Pakistan as the prominent nations in terms of contribution. Several other countries have made significant contributions, including Brazil in South America and Spain in Europe. And a few researches about ontology come from China, Canada, and Egypt. This fact also indicates that the role of ontology in software development also appeals to researchers' eyes the world over.

Figure 5 above shows the average and total of citations for all 30 Publication venues containing the papers we have amassed in the previous phase. Among these thirty venues, several journals and conferences have high citations compared with others, such as Expert System with Applications, Advanced Engineering Informatics, Journal of Systems Architecture and Information Systems. On the other hand, a high number of citations of a specific publication venue does not always indicate many papers published inside it, but a good quality paper have been published inside it and contributes a higher number of citations.

We discovered that most of the primary studies are investigative. Thirty-four publication, from total of forty-one publication in this category (83

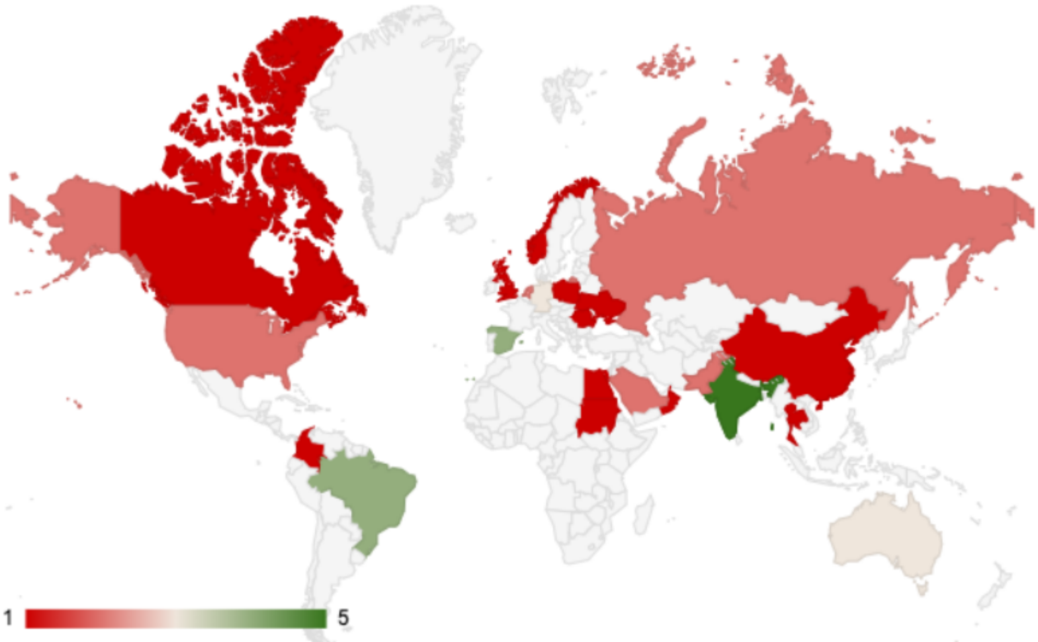

FIGURE 4 The authorship distribution of the selected studies.

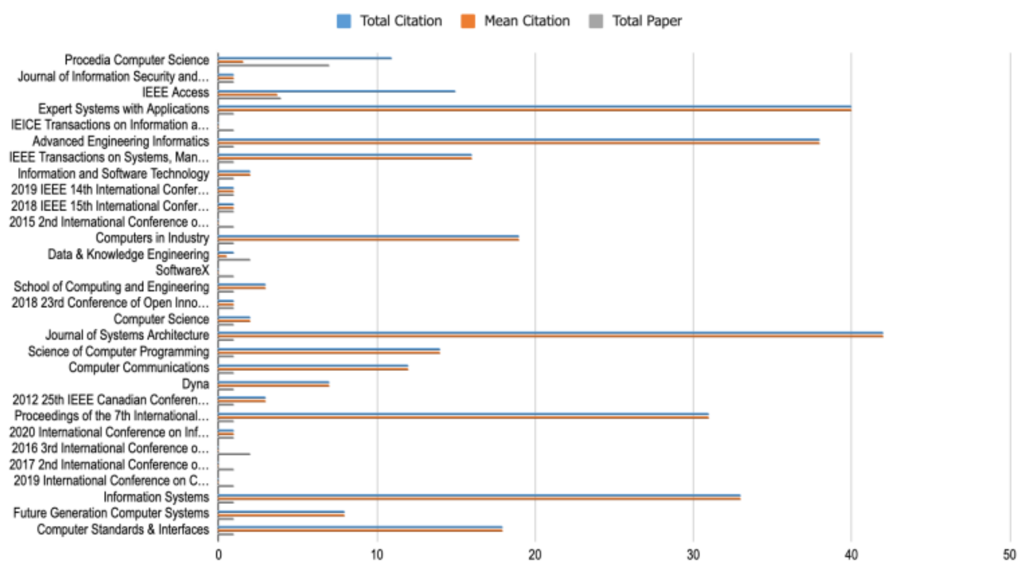

FIGURE 5 The mean and total citation for each publication venue.

There are also other research methods which were widely employed by researchers in their experimental researches. The methods include case study (1) (for analyzing data, researcher used a set of systematical procedures are used to develop deductively a theory based on certain phenomenon for data analysis), experimental report (1) (where researchers employed elicitation techniques as sub-research methods), observation (1), and survey (1). In this study, we have created a list of research methods recounted by the researchers of selected publications. Furthermore, in we also found that there is a study that evaluated more than one case

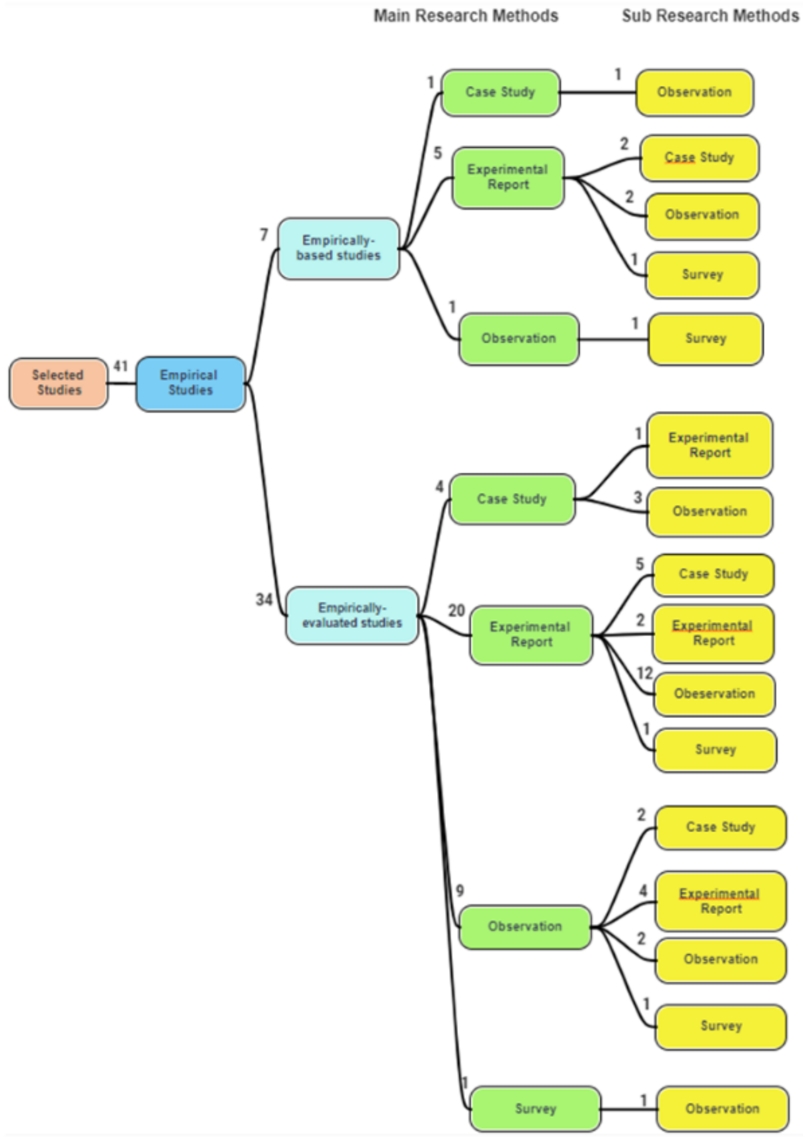

FIGURE 6 The taxonomy of the research method from the research study included.

studies in their implementation. The goal is to improve the generalizability of their method. Figure 6 dislays the taxonomy of the research methods pertinent to the role of ontology in modeling knowledge in software engineering process.

4.2 | RQ1] What is the role of ontology in software modeling?

This part describes the all kinds of application of ontology covered by all selected studies. These practices could be accepted as the vital role of ontology in modeling knowledge in software engineering activities. We identified 26 studies that deal with the application of ontology in software engineering activities. From these studies, we generalized them into ten different applications or practices. For each practice, we identified its respective potential challenges. We have to emphasize that the following grouping is inclusive. By saying this, we indicate that the current classifications only reflect the selected publications resulted from the process mentioned in section 3. The classification may also be change in the future. The occurrence off the publications is presented in Table 7.

1. Modeling the data in software development. In software development, the problem that many software developers face is changes in requirements. Many researchers use ontology to model the data so that it can resolve the ambiguity in software development. Valiente et al. [11] use ontology to describe the main elements of mapping when building the ITSM.

2. Database and application integration. Asfand-E-Yar and Ali [3] explain that ontology can be used when dealing with extensive data management to solve the heterogeneity problem in databases. The researchers explain that ontology has two different phases, to translate the database rule according to the ontology rule and find the abstract of the ontology model. Many researchers use ontology to integrate with scrum (agile software development) to build the framework to combine data, semantics, and services.

3. Describe the knowledge using a common vocabulary. The ontology was used to describe the relationship between classes [4,12]. Their research explains the same idea that ontology can be used to describe the relationship between class and concept.

4. Maintenance software testing: to reduce the conceptual gap and close the cycle between conceptual abstraction and model [13].

5. Resolving the requirement change in software development: can improve the RCM processes, increase the reliability of the changes and reduce the time consume for dealing with semantically change requests in software development [14].

6. Real-time and dynamic ontology modeling of the IoT system. The ontology was used as a modeling in the real-time and dynamic IoT system [15].

7. Quality assessment and continuous improvement. Roldán-Molina et al. [16] explain that ontology can improve the quality assessment and continuous improvement using the Deming cycle (PDCA: Plan, Do, Check, Act).

8. Standardize for ISO Software engineering pattern. Gonzalez-Perez et al. [17] explain harmonization of standards by use of a domain ontology has been advocated.They recommend the creation of a sole supporting generalization of ontology area, generated from current ISO/IEC standards, which include ISO/IEC 24744 and 24765, and accompanied by supplementary resources approved by SC7.

9. Techniques for detecting changes to software systems.

10. Application pattern in software development.

4.3 | [RQ2] What Are the Challenges of Ontology in Software Engineering?

The challenge in software development always grows over time, and it is related to the role of ontology in software development. Many researchers and software engineers have been keen to understand the importance of ontology in software engineering. How ontology and software development are related, the investigation of these researchers is focused on the interconnection between these ontology and software engineering.

1. Improvement in Software Development. The requirement change request in software development can consume time and deal with semantically wrong change requests [14]. The solution is to develop a multilevel ontology framework .

2. Knowledge abstraction [11]. The heterogeneous data and rapid change in software requirements [14] can make an impact on the limited understanding of software requirements [11], and it can be handled by developing a collaborative framework ontology and scrum agile software engineering [22].

3. Limited knowledge in malware detection [12]. The android OS version introduces runtime permission to user privacy, so it needs more time to detect and analyze the behavior and makes the problem consume more time for mobile detection malware.

4. Evaluation when integrating ontology with others [4]. The evaluation of ontology with other applications involved in two organizations. The problem is limited to the integration solution produced and can be handled by exploring other SEON ontologies to provide the integrated data .

TABLE 6 Summary of Practices and The Respected Studies has been investigated them.

Role of Ontology	Freq	Study
Modeling the Data in software development	5	Bhatia et al. [2], Valiente et al. [11], Puchianu and Bautu [18], Rocha et al. [19], Beydoun et al. [20]
Database and application integration	5	Asfand-E-Yar and Ali [3], Junior et al. [4], Petnga and Austin [21], Takhom et al. [22], Adnan and Afzal [23]
Describe the knowledge using common vocabulary	6	Asfand-E-Yar and Ali [3], Junior et al. [4], Jannath and S [12], Petnga and Austin [21], Takhom et al. [22], Adnan and Afzal [23]
To Resolve Requirement change in Software Development	3	Alsanad et al. [14], Gregorio et al. [24]
Maintenance software testing	2	Popereshnyak and Vecherkovskaya [1], García-Peñalvo et al. [13]
Building the concept of software quality	1	Alsanad et al. [14]
Developing systems in data integration	1	John et al. [25]
Application pattern in software development	1	Deb et al. [26]
Software development model	1	De Graaf et al. [27]
Techniques for detecting changes to software systems	1	Peldszus et al. [28]
Software development process	1	Abdalazeim and Meziane [29]
Software data model	1	Dahling et al. [30]
Software development data model process	1	Olszewska and Allison [31]
Modeling software knowledge	1	Wen and Katt [32]
Tools for building software	1	Stadnicki et al. [33]
Technical model software	1	Zou et al. [34]
Software engineering process	1	Murtazina and Avdeenko [35]
Software development process	2	Wongthongtham et al. [36], Ortega-Ordoñez et al. [37]
Software measurement & systems Integration	1	Fonseca et al. [38]
Real-time and dynamic ontology modeling of the IoT system	1	Chen et al. [15]
Software engineering process	3	Orellana and Mandrick [39], Wiebe and Chan [40], Van Kervel et al. [41]
Analyzing social media Big Data in Cloud	1	Chauhan et al. [42]
Quality assessment & continuous improvement	1	Roldán-Molina et al. [16]
Support of requirements engineering in agile software development	1	Murtazina and Avdeenko [43]
Reverse engineering of conventional software	1	Bhatia et al. [2]
Service-oriented system	1	Shen et al. [44]
ISO Software engineering standards	1	Gonzalez-Perez et al. [17]

5. Management process in software development.

6. Absence of effective utilization of software development. A method in ontology manufacturing is comprised of a set of method, technique, process, and activity. On the other hand, ineffective methodologies quality for support ontology development needs the solution of a new software-centric approach to develop ontology based on a well-defined Ontology Development Life Cycle (ODLC).

7. Ontology in the context of software security. Ontology can be utilized to solemnize the knowledge of the security context required to realize a set o security requirements.

8. Understanding why a development group or a software engineer does not stick to the predefined project plan. There are unsuccessful software engineering cases when the development group or a software engineer does not stick to the predefined project plan. Theses cases tend to produce a software product which unsuitable to its requirements. This can lead to the problem of software is not suitable for the requirements, and the stakeholders or customers are not satisfied. The solution would be coupling ontology with a multi-agents system. This could lead to a greater effortlessness of communication by combining the approved ontology system knowledge, the domain knowledge, and the principles and best-practices of software engineering practices into a publicly accessible resource platform and let the resources to be accessible by all member of the teams within the organization.

TABLE 7 Challenges of ontology in software engineering.

Challenge	Description	Impact	Solution
Improvement in Software Development	the requirement change request in software development	time consumed and dealing with semantically wrong change requests [14]	Developing a multilevel ontology framework [14]
Knowledge abstraction [11]	The heterogeneous data and rapid change in software requirement [14]	Limited understanding of software requirements [11]	Developing a collaborative framework ontology and scrum agile software engineering [22].
Limited knowledge of malware detection [12]	The android OS version introduces runtime permission to user privacy, so it needs more time to detect and analyze the malware behavior	Consume more time detecting mobile malware	
Evaluation when integrating ontology with others [4]	The evaluation ontology with other applications involved in two organization	Limited the integration solution that has been produced	Exploring other SEON ontology to provide the integrated data [4]
Management process in software development			
Absence of effective utilization of the software development	A methodology in ontology engineering is composed of methods, techniques, processes, and activities	Ineffective methodologies quality for support ontology development	A new software-centric approach to ontology development with a defined Ontology Development Life Cycle (ODLC)
Ontology in the context of software security	Ontologies can be used to formalize the knowledge of security contexts required for implementing security requirements.	Subject for continuous change, which demands consistent evolution of ontologies	Able to detect patterns in semantics, which can be specifically defined using graphs transformation rules, but does not rely on information about processes such as continuously managed changelog.
Understanding why a team or a software engineer does not follow the project plan	Several cases happen in engineering when the team or software engineer does not follow the project plan and the software is not suitable for the requirements.	The software is not suitable for the requirements, and the stakeholders or customers are not satisfied.	Ontologies coupled with a multi-agent system allow greater ease of communication by aggregating the agreed knowledge about the project, the domain knowledge, and the concepts of software engineering into a shared information resource platform and allow them to be shared among the distributed teams across the sites.

5 | RESULTS AND DISCUSSION

An interesting information emphasized in the evaluation of the 41 selected publication is the topographical location of the researchers. The observation shows that many of the researchers are affiliated to South Asia (e.g. India and Pakistan). Meanwhile, the USA and European nations generated the smallest amount of researches with respect to the role of ontology in software development. The only contribution from Southeast Asian countries is Thailand and East Asian countries. The only contribution is from China.

We also observed that great number of publications mentioned in this methodical assessment have utilized an experimental-based approach. Based on this findings, we can established that researchers have acknowledged the role of ontology in software development and do not reproduce it in the classroom. 59% of studies speak of the experimental report examples. Furthermore, there is also study that experimented with more than one case studies on the role of ontology in modeling knowledge in software engineering. This effort indicate that generalizability ontology in software engineering activities could be realized. Thus, effort is still need to scale up the bigger problems. From all the case studies, we found five case studies where the sub-research method was never described explicitly. The finding denotes to the issue of clarity in presenting the result of the researches. Thirty-four publications were classified as exploratory based on empirical investigation. There are also afresh recommended solutions from previous studies. These findings reveal that a greater part of the attempts are still concentrating on comprehending, identifying, and implementing the role of ontology in software development, ontology challenges, and issues.

Additionally, we also have acknowledged the existing practices of ontology's role in software development in RQ1, which confirms the role of ontology in modeling knowledge in software development. However, we may not be able to constantly track the findings to specific ontology engineering contexts. More experimental studies are needed to realize the applications and implementation of the findings and their contribution to the current body of knowledge in the domain of software engineering.

We have identified the challenge of Ontology in response to the RQ2. The role of Ontology has several challenges, such as knowledge abstraction, changes in software requirements in software development, and limited knowledge of mobile malware detection. The ontology practices introduce several steps to resolve and deal with the challenges. However, matching the list of 22 papers that answer the RQ1 with the list of challenges that ontology help to solve (answer to RQ2). However, in answering the RQ2, we found several studies not answering the challenge of ontology in software development. This is an inconsistency and needs further investigation. Likewise, the role of ontology is to describe the concept and relation between classes and knowledge representation [22].

Overall, it was evident that ontology's central role in software development is to model the data and describe the concept between classes and knowledge. It can help the software developers to reduce the time consumed when dealing with the semantically wrong change request. In integration with other methods and applications(example: ontology and scrum), contributed to providing data and improving estimated time and people allocation. Therefore more research should be conducted to cover the gap in this research.

6 | CONCLUSION

This paper describes a methodical literature assessment on practices and the vital role of ontology in software development. This assessment was carried out by adopting available procedures and steps to develop a literature review to collect, extract, and categorize all current and available electronic publications which discuss about the vital role of ontology in modeling knowledge in software development. The initial hundred and fifty publications were collected from well-known electronics journal databases. Another forty-one publications were found by scanning relevant publications listed in the reference of those initial publications and papers that cited them. All collected publications were evaluated for the quality of the results that they have produced. Selected articles were later examined, investigated, and categorized into the a thematic group based on the research questions:

- The reviews show that Ontology has the primary role as a tool to represent knowledge and describe the relationship between classes and concepts.
- The finding of our research provides future dimensions for practitioners that work in software development. We have determined that further research on the ontology in modeling knowledge in software development and its implementation in real-world applications. Our reviews on ontology's role in software engineering are open to be evaluated and improved by adding more relevant studies and researchers in the future.

6.1 | Implications of The Study

This methodical literature assessment has several scientific and practical implications. For researchers, the studies indicate that extensive efforts are needed to conduct scalable experimental studies on real-world cases. These efforts would provide more information and assurance to software developers in order to understand the role of ontology in modeling knowledge in software development processes. Thus there is a need to integrate ontology with other methods and applications in several domains and organizations. However, sharing the ontology software development in Software engineering communities is required to update the ontology role and the usage so that ontology development can proliferate.

6.2 | Limitations of The Study

The fundamental limitation of any methodical literature assessment may be produced during these two processes. First, the study selection process may contain bias. There should be a mechanism to test whether the keyterms used in search criteria indeed accurate and represent the main topic. Second, the likely inexactness in publication collection process due to the varying of electronic databases. We carried out the subsequent actions as for designing the research strategy to remove the potential bias and guarantee precision and accuracy in the publication selection process. The first action is to treat the keyword used for searching process as a refinement procedure that includes further investigation. We parsed our research questions to determine the keywords for extensive search on the electronic databases. The second action is encompassing solely publications that mainly focus on the role of ontology in software development and not in other studies. To reduce bias by the personal preferences, the articles have been examined by the two researchers . Additionally, we also recognized that numerous publications have insufficient information in order to be included in this literature assessment. To be more précised, we discovered that the degree of specificity where the research methods were explained in 22 publications.

Moreover, we acknowledged that in every publication where challenges are mentioned (challenges of ontology in software development), the underlying causes of the challenge didn't discuss thoroughly. It is very difficult to modify the state of affairs. This is because it is assumed that the some researchers may have selected the stated challenges for particular rationales. Nonetheless, the researchers have concentrated on particular research objectives and research questions. Additionally, the motivating causes for the challenges may have been ignored.

CREDIT

Marco Ariano Kristyanto: Conceptualization, Methodology, Data Curation, Writing - Original Draft, and Investigation. **Ravi Vendra Rishika:** Conceptualization, Methodology, Data Curation, Writing - Original Draft, and Investigation. **Franky Rawung:** Conceptualization, Methodology, Data Curation, Writing - Original Draft, and Investigation.**Evi Triandini:** Supervision, Validation, Writing - Review & Editing.

References

1. Popereshnyak S, Vecherkovskaya A. Modeling Ontologies in Software Testing; Modeling Ontologies in Software Testing. In: International Scientific and Technical Conference on Computer Sciences and Information Technologies IEEE; 2019. p. 236–239.

2. Bhatia M, Kumar A, Beniwal R. Ontology-Based Framework for Reverse Engineering of Conventional Software. In: Proceedings of the 10th INDIACom; 2016 3rd International Conference on Computing for Sustainable Global Development, INDIACom 2016 IEEE; 2016. p. 3645–3648.

3. Asfand-E-Yar M, Ali R. Semantic integration of heterogeneous databases of same domain using ontology. IEEE Access 2020;8:77903–77919.

4. Junior PSS, Barcellos MP, de Almeida Falbo R, Almeida JPA. From a scrum reference ontology to the integration of applications for data-driven software development. Information and Software Technology 2021;136:106570.

5. Isotani S, Ibert Bittencourt I, Francine Barbosa E, Dermeval D, Oscar Araujo Paiva R. Ontology Driven Software Engineering: A Review of Challenges and Opportunities. IEEE Latin America Transactions 2015;13(3):863 – 869.

6. Tebes G, Peppino D, Becker P, Matturro G, Solari M, Olsina L. Analyzing and documenting the systematic review results of software testing ontologies. Information and Software Technology 2020;123:106298. https://www.sciencedirect.com/science/article/pii/S0950584920300495

7. Mkhinini MM, Labbani-Narsis O, Nicolle C. Combining UML and ontology: An exploratory survey. Computer Science Review 2020;35(1):1–14.

8. Verdonck M, Gailly F, De Cesare S, Poels G. Ontology-driven conceptual modeling: A'systematic literature mapping and review. Applied Ontology 2015;10(3-4):197 – 227.

9. Kitchenham B, Pretorius R, Budgen D, Brereton OP, Turner M, Niazi M, et al. Systematic literature reviews in software engineering-A tertiary study. Information and Software Technology 2010;52(8):792–805.

10. Inayat I, Salim SS, Marczak S, Daneva M, Shamshirband S. A systematic literature review on agile requirements engineering practices and challenges. Computers in Human Behavior 2015 10;51:915–929.

11. Valiente MC, García-Barriocanal E, Sicilia MÁ. Applying ontology-based models for supporting integrated software development and IT service management processes. IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews 2012;42(1):61 – 74.

12. Jannath JN, S MSB. Detection of malicious android applications using ontology-based intelligent model in mobile cloud environment. Journal of Information Security and Applications 2021 5;58:1–15.

13. García-Peñalvo FJ, Colomo-Palacios R, García J, Therón R. Towards an ontology modeling tool. A validation in software engineering scenarios. Expert Systems with Applications 2012;39(13):11468–11478.

14. Alsanad AA, Chikh A, Mirza A. Multilevel ontology framework for improving requirements change management in global software development. IEEE Access 2019;7:71804–71812.

15. Chen G, Jiang T, Wang M, Tang X, Ji W. Modeling and reasoning of IoT architecture in semantic ontology dimension. Computer Communications 2020 3;153:580–594.

16. Roldán-Molina GR, Ruano-Ordás D, Basto-Fernandes V, Méndez JR. An ontology knowledge inspection methodology for quality assessment and continuous improvement. Data and Knowledge Engineering 2021;133(May):1–16.

17. Gonzalez-Perez C, Henderson-Sellers B, McBride T, Low GC, Larrucea X. An Ontology for ISO software engineering standards: 2) Proof of concept and application. Computer Standards and Interfaces 2016;48(Nov):112–123.

18. Puchianu CM, Bautu E. Conceptual and ontological modeling of in-vehicle life-logging software systems. Procedia Computer Science 2020;176:2635–2644.

19. Rocha R, Araújo A, Cordeiro D, Ximenes A, Teixeira J, Silva G, et al. DKDOnto: An ontology to support software development with distributed teams. Procedia Computer Science 2018;126:373–382.

20. Beydoun G, Low G, Garcia-Sanchez F, Valencia-Garcia R, Martínez-Bejar R. Identification of ontologies to support information systems development. Information Systems 2014;46:45–60.

21. Petnga L, Austin M. An ontological framework for knowledge modeling and decision support in cyber-physical systems. Advanced Engineering Informatics 2016 1;30:77–94.

22. Takhom A, Usanavasin S, Supnithi T, Boonkwan P. A collaborative framework supporting ontology development based on agile and scrum model. IEICE Transactions on Information and Systems 2020 12;E103D:2568–2577.

23. Adnan M, Afzal M. Ontology based multiagent effort estimation system for scrum agile method. IEEE Access 2017 11;5:25993–26005.

24. Gregorio JL, Oliveira HCD, Figueiredo LR, Prado SG. Specification of Software Requirements with the Support of Business Process Ontologies. In: CITS 2019 - Proceeding of the 2019 International Conference on Computer, Information and Telecommunication Systems IEEE; 2019. p. 1–5.

25. John S, Shah N, Stewart C. Towards a Software Centric Approach for Ontology Development: Novel Methodology and its Application. In: Proceedings - 2018 IEEE 15th International Conference on e-Business Engineering, ICEBE 2018 Institute of Electrical and Electronics Engineers Inc.; 2018. p. 139–146.

26. Deb CK, Marwaha S, Malhotra PK, Wahi SD, Pandey RN. Strengthening soil taxonomy ontology software for description and classification of USDA soil taxonomy up to soil series. In: 2015 International Conference on Computing for Sustainable Global Development, INDIACom 2015; 2015. p. 1180–1184.

27. De Graaf KA, Liang P, Tang A, Van Hage WR, Van Vliet H. An exploratory study on ontology engineering for software architecture documentation. Computers in Industry 2014;65(7):1053–1064.

28. Peldszus S, Burger J, Kehrer T, Jurjens J. Ontology-driven evolution of software security. Data and Knowledge Engineering 2021 7;134:1–25.

29. Abdalazeim A, Meziane F. A review of the Generation of Requirements Specification in Natural Language Using Objects UML Models and Domain Ontology. In: Procedia CIRP, vol. 189 Elsevier B.V.; 2021. p. 328–334.

30. Dahling S, Razik L, Monti A. OWL2Go: Auto-generation of Go data models for OWL ontologies with integrated serialization and deserialization functionality. SoftwareX 2020 7;12:1–9.

31. Olszewska JI, Allison IK. ODYSSEY: Software development life cycle ontology. In: IC3K 2018 - Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, vol. 2 SciTePress; 2018. p. 303–311.

32. Wen SF, Katt B. An Ontology-Based Context Model for Managing Security Knowledge in Software Development. In: 23rd Conference of Open Innovations Association (FRUCT) IEEE; 2018. p. 416–424.

33. Stadnicki A, Pietron F, Burek P. Towards a modern ontology development environment. Procedia Computer Science 2020;176:753–762.

34. Zou M, Basirati MR, Bauer H, Kattner N, Reinhart G, Lindemann U, et al. Facilitating consistency of business model and technical models in product-service-systems development: An ontology approach. IFAC-PapersOnLine 2019 9;52:1229–1235.

35. Murtazina MS, Avdeenko TV. An ontology-based approach to support requirements traceability in agile development. Procedia Computer Science 2019;150:628–635.

36. Wongthongtham P, Chang E, Dillon TS, Sommerville I. Ontology-based multi-site software development methodology and tools. Journal of Systems Architecture 2006 11;52:640–653.

37. Ortega-Ordoñez WA, Pardo-Calvache CJ, Pino-Correa FJ. Ontoagile: An ontology for agile software development processes. DYNA (Colombia) 2019;86(209):79–90.

38. Fonseca VS, Barcellos MP, de Almeida Falbo R. An ontology-based approach for integrating tools supporting the software measurement process. Science of Computer Programming 2017 2;135:20–44.

39. Orellana D, Mandrick W. The ontology of systems engineering: towards a computational digital engineering semantic framework. Procedia Computer Science 2019;153:268–276.

40. Wiebe AJ, Chan CW. Ontology Driven Software Engineering. In: 2012 25th IEEE Canadian Conference on Electrical and Computer Engineering: Vision for a Greener Future, CCECE 2012; 2012. p. 1–4.

41. Van Kervel SJH, Dietz JLG, Hintzen J, Van Meeuwen T, Zijlstra B. Enterprise Ontology driven software engineering. In: ICSOFT 2012 - Proceedings of the 7th International Conference on Software Paradigm Trends; 2012. p. 205–210.

42. Chauhan A, Vijayakumar V, Vincent R, Pradeep KV. Towards the development of a framework for socially responsible software by analyzing social media big data on cloud through ontological engineering. Procedia Computer Science 2015;50:524–530.

43. Murtazina M, Avdeenko T. The Ontology-Driven Approach to Intelligent Support of Requirements Engineering in Agile Software Development. In: Proceedings of ITNT 2020 - 6th IEEE International Conference on Information Technology and Nanotechnology Institute of Electrical and Electronics Engineers Inc.; 2020. p. 1–6.

44. Shen J, Beydoun G, Low G, Wang L. Aligning ontology-based development with service oriented systems. Future Generation Computer Systems 2014 3;32:263–273.

How to cite this article: Triandini E., Kristyanto M.A., Rishika R.V., Rawung F. (2021), A Systematic Literature Review of The Role of Ontology in Modeling Knowledge in Software Development Processes, *IPTEK The Journal of Technology and Science*, *32(3):159-175*.

A SYSTEMATIC LITERATURE REVIEW OF THE ROLE OF ONTOLOGY IN MODELING KNOWLEDGE IN SOFTWARE DEVELOPMENT PROCESSES

ORIGINALITY REPORT

9%
SIMILARITY INDEX

PRIMARY SOURCES

1	scholar.archive.org Internet	150 words — 2%
2	studentsrepo.um.edu.my Internet	118 words — 2%
3	espace.curtin.edu.au Internet	86 words — 1%
4	repository.poliban.ac.id Internet	84 words — 1%
5	Kaiss Elghariani, Nazri Kama. "Review on Agile requirements engineering challenges", 2016 3rd International Conference on Computer and Information Sciences (ICCOINS), 2016 Crossref	66 words — 1%
6	garuda.kemdikbud.go.id Internet	45 words — 1%
7	www.x-mol.com Internet	44 words — 1%
8	"Enterprise Information Systems", Springer Science and Business Media LLC, 2021	37 words — 1%

Crossref

EXCLUDE QUOTES	ON	EXCLUDE SOURCES	< 1%
EXCLUDE BIBLIOGRAPHY	ON	EXCLUDE MATCHES	OFF