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

ORIGINAL RESEARCH

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

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

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

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

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

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

Ontologies in software development are explained as showing the properties of a subject area and how they are related to each other by defining a set of concepts and categories that represent the subject. It is used to determine the ambiguity in the software requirements specification. Although claimed to be beneficial, the software development 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. Our goal is to map and explain the evidence 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 conducted a systematic review of the literature published between 2012 and 2021 and identified 150 papers that discuss the role of ontology in modeling knowledge in software development processes. We formulated and applied specific inclusion and exclusion criteria in two rounds to determine the most relevant studies for our research goal. The review identified 22 practices that explain ontologies' primary role in software development processes. However, our findings suggest ontology's role in software engineering as a research context needs additional attention. A more empirical result is required to understand better the role of ontology in modeling knowledge in software development with non-functional requirements and self-organizing teams.

KEYWORDS:

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

1 | INTRODUCTION

In computer science, an ontology phrase is a data model representing a set of concepts within a domain and the relationship between those concepts. 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 modeling is divided into two phases: initially translating the database rules according to ontology rules and finding an abstract of the ontology model [3]. The need for complex systems and software always grows. However, the risk involved in the development of software also increases. The risk of software failure, unsuitable requirements, and defects as occurred in software engineering activities. The problem is that each reference model employs its vocabulary to deal with this phenomenon [4].

This paper conducts a systematic literature review to help researchers solve the problem. The article's structure is as follows: Section 2 discusses the previous literature review on the role of ontology in software development processes and identifies the gap in the literature and a need for a deeper investigation. Section 3 presents our research questions, and the method followed for the current services in The role of Ontology in software modeling. Section 4 summarizes the key finding of our study. Section 5 provides the discussion of the result. Section 6 concludes the article, provides implications for researchers and industry practitioners, and defines the study's limitations.

2 | PREVIOUS RESEARCHES

In the software engineering research literature, there are a few examples of reviews on 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 systematic literature review on the use of ontologies and techniques benefit in the Software Engineering area. The study focused on usability issues in the ontology model concerning design. The findings show that 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] conducted a systematic literature review to explain software testing ontologies. It revealed that the software testing ontology minimizes the current heterogeneity, ambiguity, and incompleteness problem in properties and relationships. Tebes finding none of the conceptualized ontologies are directly linked with Non-Functional Requirements (NFRs) and Functional Requirements (FRs) components.

Mkhinini et al. [7] conducted a systematic literature review on combining UML and ontology. The study investigates the underlying paradigm, and the approaches to combining these two are increasingly emerging. The paper aims to provide a comprehensive overview of both domains by conducting a literature review of the relevant research work. They also created a survey to find the relationship between UML and ontology from theoretical and practical perspectives. They present a detailed classification of the existing work based on considered issues and practical use cases.

Verdonck et al. [8] examined Ontology-Driven Conceptual Modeling (ODCM) in software engineering. The literature review contributed to analyzing the nature of the research contributions and establishing its current state of the art by positioning, evaluating, and interpreting relevant research to date related to ODCM.

3 | RESEARCH METHODOLOGIES

We follow the guidelines proposed by Kitchenham et al. [9] in the research process. Subsequently, we presented our main step in our systematic review: planning, conducting, and reporting the review 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

We planned this review by proposing research questions relevant to our objectives. Defining the search strategy, search string, and inclusion/exclusion criteria. We present these below.

3.1.1 | Review objectives and research questions

With the increased use of ontology in software development, it is most important to study the role of ontology in software development. As defined in Software development, ontology modeling is divided into two phases: initially translating the database rules according to ontology rules and finding an abstract of the ontology model[3]. Therefore the main goal of this research is to understand the role 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.

To fulfill these objectives, we formulated the research question. We formulated the following 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 defining our research questions and goals, we started with the formulation of a formal search strategy to analyze all available empirical-specific materials for the objective of this research. The detailed plan in this research on defining the search space, which included electronic databases, journals, and printed proceedings, is shown in Table 2. The studies we did were initially started by retrieving from the electronic databases, finding the related publications by using Snowballing (Forward and Backward) method, and analyzing to identify 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. Then, the inclusion and exclusion criteria were also applied to the retrieved studies using two rounds and involving several researchers, as explained in Section 3.1.4.

3.1.3 | Search Criteria

The search criteria used in the phase of this review consist of two parts defined 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 composed the search string or keyword in each electronic database manually based on the search functionality offered by each electronic database. We thought and treated the search process in each database as a process of learning and experimentation for our review.

3.1.4 | Inclusion and Exclusion Criteria

To determine whether a study should be included, the following inclusion and exclusion criteria were used:

- **Inclusion criteria:** (I1) The study is a peer-reviewed publication; (I2) The study is in English; (I3) It is relevant to the search terms defined in Section 3. 1.3; (I4) It is an empirical research paper, an experience report, or workshop paper; and (I5) This research is published between January 2012 and July 2021.

- **Exclusion criteria:** (E1) Studies that do not focus explicitly on ontology methods but only refer to ontology software development methods as a side topic (e.g., studies that cite ontology as an adjective; (E2) Studies that do not discuss the role of Ontology in software development; (E3) Studies that do not meet inclusion criteria; and (E4) Opinion, viewpoint, keynote, discussion, editorials, comments, tutorials, prefaces, and anecdote papers and presentations in the slide formats without any associated papers.

3.2 | Conducting the Review

This section presents our findings from our extraction process and searches from relevant sources and databases.

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 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

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**

TABLE 4 Quality Criteria for study selection.

Quality Criteria from Obtained Studies	Response Grading	Grade
c1(is the research aim/objective clearly defined?	(1,0.5,0)(yes,nominally no,no)	30 studies, 66.67%
c2(is the context research well addressed)?	(1,0.5,0)(yes,nominally no,no)	30 studies , 66.67%
c3(Are the finding clearly stated ?)	(1,0.5,0)(yes,nominally no,no)	21 studies , 46,67%
c4(based on the finding how valuable the research?)	>80% =1, <20% =0, in between = 0.5	

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. The independent quality assessments were conducted for the 45 studies, as explained in Section 3.1.4, and disagreements were resolved through a discussion.

3.2.3 | Methodological Quality Assessment

These systematic reviews use the quality criteria based on Inayat et al.[10] to assess methodological quality studies selected for the review: the criteria were later used to assess the quality of empirical studies discussing the role of Ontology in software development methods. These quality criteria (presented in Table 4) comprise questions that measure the extent to which a study is satisfactory and will contribute to the scope of the investigation. The criteria cover the thoroughness, trustworthiness, and significance of the studies. We choose these criteria because they can be used to investigate the usefulness of synthesis findings and the result of interpretation Kitchenham et al.[9]. Several recent systematic reviews previously employed the quality measure associated with these criteria.

We evaluated each study based on the criteria presented in Table 4 for categorization and rating. For a better categorization and rating of the studies, we utilized an ordinal scale based on our quality assessment criteria (Table 4) instead of a dichotomous scale. The first criteria (C1) involved assessing the objective of each study. This question was answered positively in 30 of each study. The second criterion (C2) assessed whether the research context was adequately addressed and described. The question was answered positively in 30 of the studies. In the third criterion (C3), we inquired about the clear statement finding of the studies. This question was answered positively in 21 of the studies. The three mentioned above researchers established the heuristic scores (quality measures/C4). The normalized score of selected studies which are based on quality scores, are shown in Figure 1 .

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.

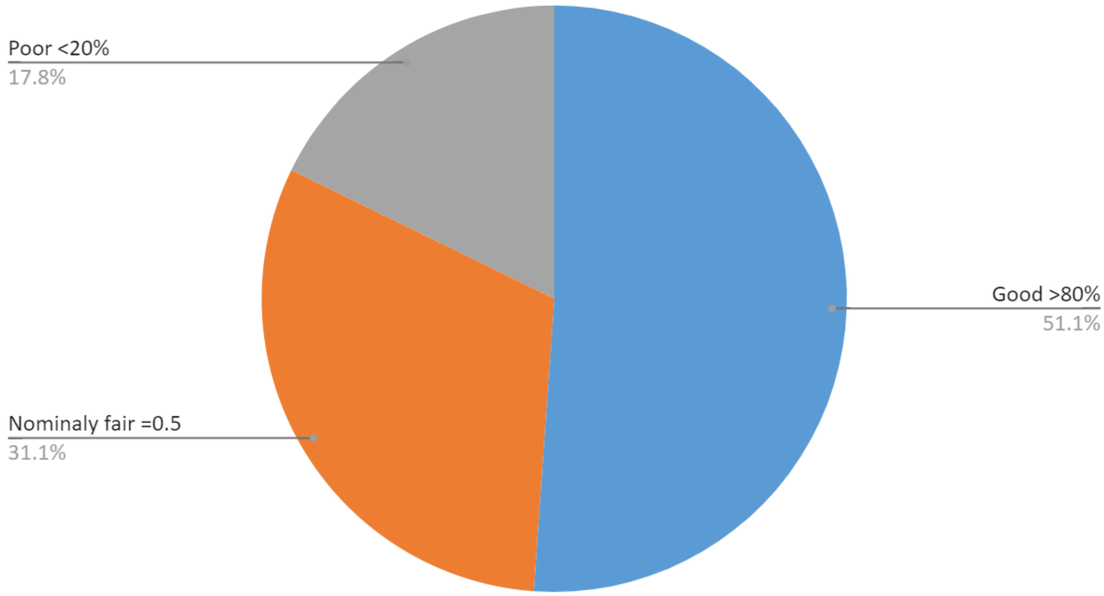

FIGURE 1 The distribution of the studies group by their quality scores (in percentage).

4 | FINDINGS OF OUR REVIEW

In this segment, we describe the findings of our critiques in light of the study's questions.

4.1 | Overview of Studies

As mentioned in the previous part, we have got recognized 41 researchers. Table 5 suggests the distribution of examination and publication sources. 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 one or papers posted in it. This means there is no single supply desired via Ontology in Software Development 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 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

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**

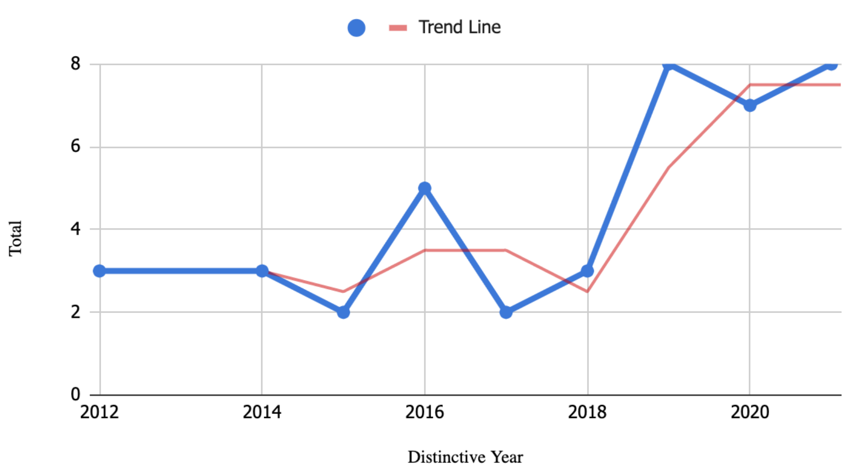

FIGURE 2 The study distribution based on its publication year.

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

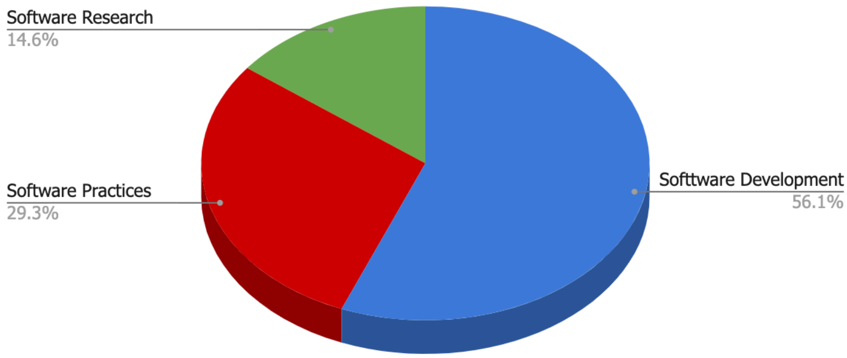

FIGURE 3 The study distribution based on their topics.

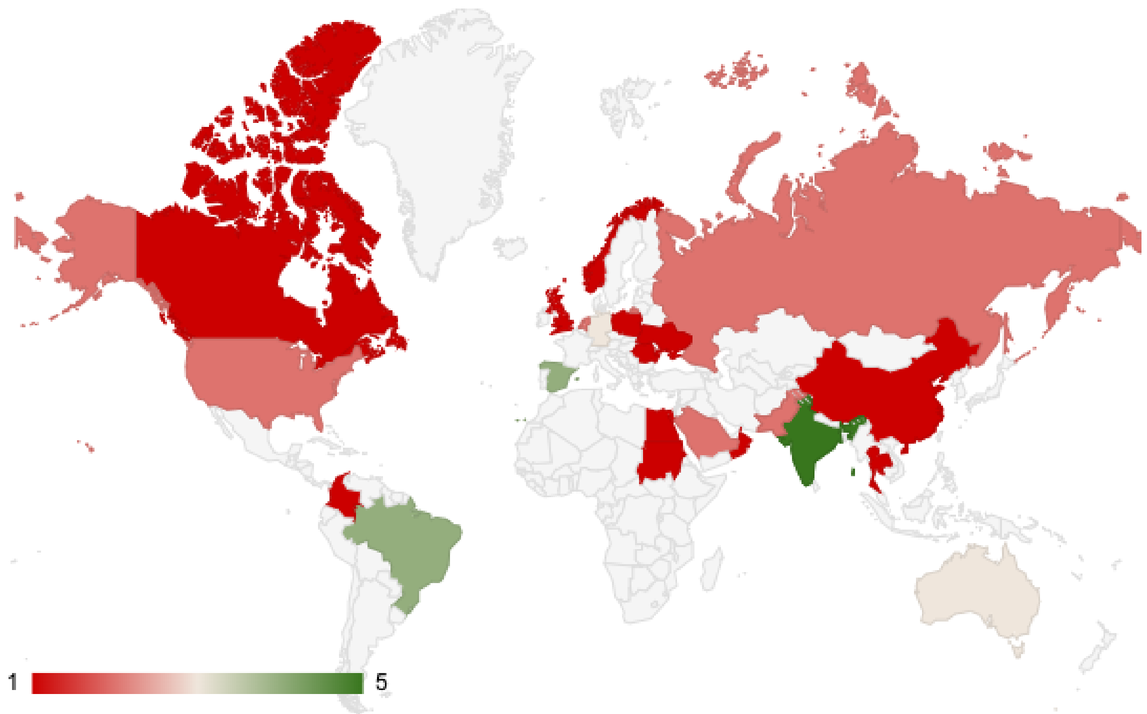

FIGURE 4 The authorship distribution of the selected studies.

role of ontology in Software Development and indicated that ontology has been applied and has a significant role in Software Development.

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.

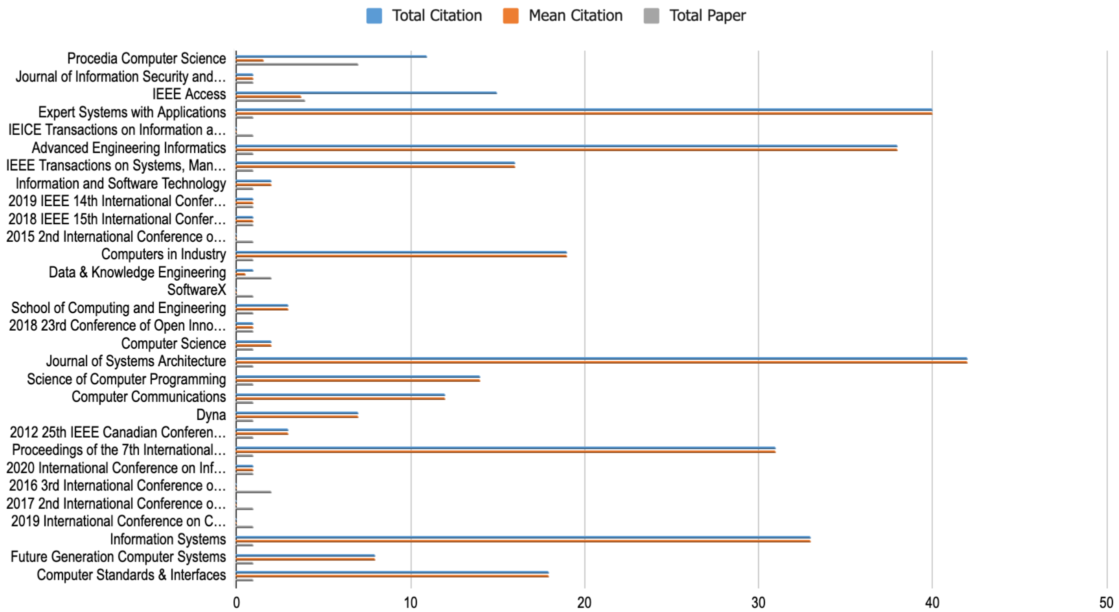

FIGURE 5 The mean and total citation for each publication venue.

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 found that majority of the studies are exploratory. Out of 41 studies in this category are empirically evaluated (34 of 41) based on the evaluation of the method or tools without any experimental or real-life investigation meanwhile the others used qualitative research methods (e.g., surveys and experimental reports) and are empirically-based (7 of 41) and are exploratory. Out of 41 studies, we found the experiments are used as methods in 61% (25 of 41) of the studies. In the other 39% of the studies within the group of 41, The method used in case studies was found to have no experimentation. For this research, we use the term a sub-research method when the research technique is applied as a part of the research process, e.g., a case study may include interviews and surveys as data collection techniques. Observation and surveys are considered sub-research methods.

Other research methods used in empirically-based studies include case study (1) (with grounded theory as a sub-method for data analysis), experimental report (1) (with interviews as a sub-method), observation (1), and survey (1). We have listed the research methods reported by the authors of selected studies, such as ethnography and observation, separately. Moreover, in one of the studies, multiple cases were used to improve the generalisability of the results. Figure 6 shows te taxonomy of the research methods relevant to the role of ontology in modeling knowledge in software development process.

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

Below We describe the ontology practices that we found to be adopted in the role of ontology in modeling knowledge in software development. For each practice, we identified its respective potential challenges. It is important to note that the list below is inclusive, i.e., it reflects what we have collectively found in 26 studies. The frequency of occurrence and the studies reporting each of the practices can be found 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.

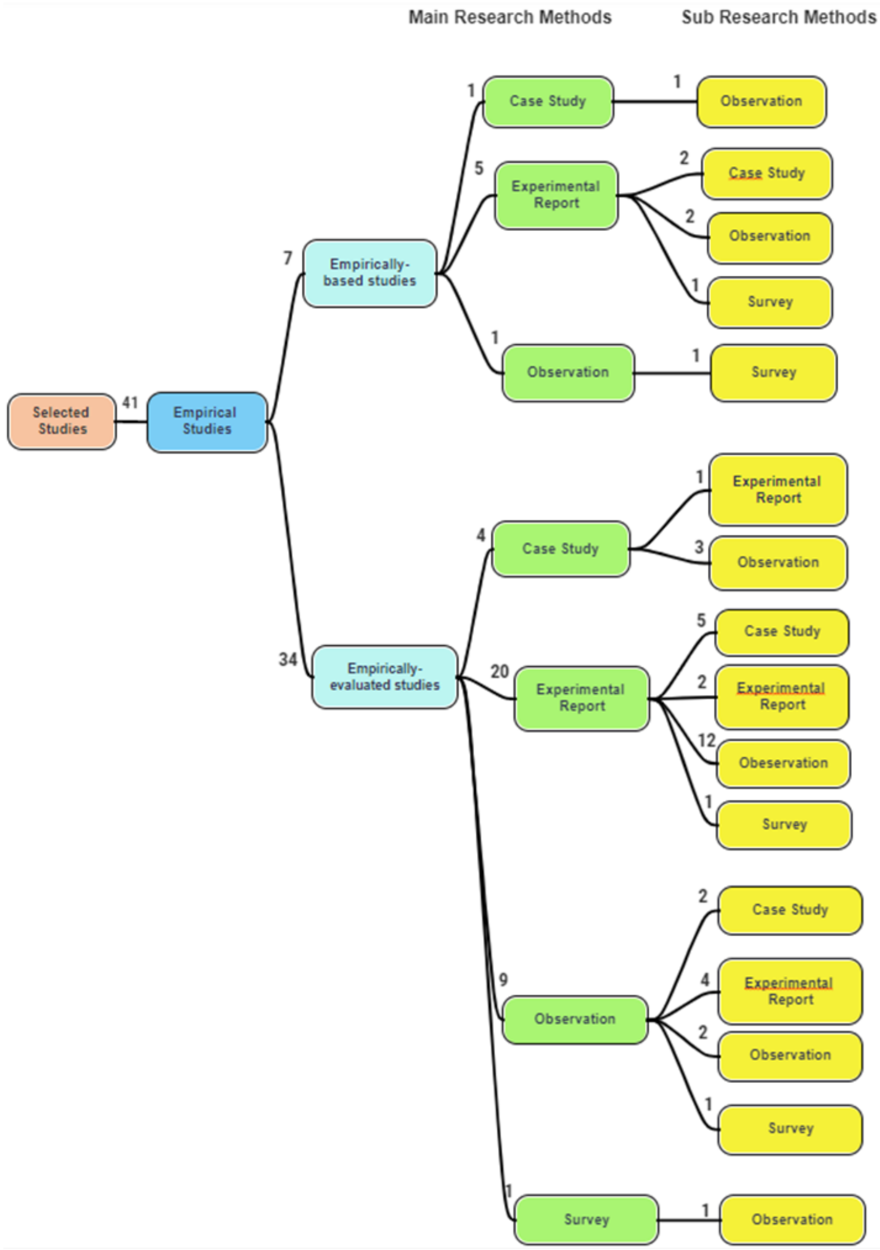

FIGURE 6 The taxonomy of the research method from the research study included.

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].

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. 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 single underpinning abstract domain ontology, created from existing ISO/IEC standards, including ISO/IEC 24744 and 24765, and supplemented by other sources authorized 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 .
5. Management process in software development.
6. Absence of effective utilization of software development. A methodology in ontology engineering is composed of methods, techniques, processes, and activities. On the other hand, ineffective methodologies quality for support ontology development needs the solution of a new software-centric approach to ontology development with a defined Ontology Development Life Cycle (ODLC).
7. Ontology in the context of software security. Ontologies can be used to formalize the knowledge of the security context needed to implement security requirements.
8. 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. This can lead to the problem of software is not suitable for the requirements, and the stakeholders or customers are not satisfied. The solution is Ontologies coupled with a multi-agents system to 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 allowing them to be shared among the distributed teams across the sites.

5 | RESULTS AND DISCUSSION

An important aspect highlighted in the analysis of our 41 selected studies is the geographic location of the authors. It is observed that many of the authors are from South Asia (e.g., India and Pakistan). Meanwhile, the USA and European countries produced the least number of studies regarding 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 found that most of the studies reported in this review have used an experimental Based approach. This established that researchers acknowledge the role of ontology in software development and do not reproduce it in the classroom. 59% of studies speak of the experimental report examples, with some of them having discussed multiple cases of the role of ontology in modeling knowledge in software development. This helps the generalisability of the result and scales up the bigger settings. In 5 case studies, the sub-research method was not mentioned explicitly. This refers to the problem of clarity in the presentation of the studies. Thirty-four of the studies were exploratory based on empirical investigation. Newly proposed ideas and literature review papers. These findings reveal that most of the efforts are focused on understanding, identifying, and implementing the role of ontology in software development, ontology challenges, and issues.

Furthermore, we have identified the current practices of ontology's role in software development in RQ1, which ensure the role of ontology in modeling knowledge in software development. However, we could not always trace our findings to particular project contexts. More empirical studies are required to implement these practices and contribute findings to the existing body of knowledge.

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.

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 presents a systematic literature review on practices in the role of ontology in software development. This review was conducted by following available guidelines to develop a literature review to search and categorize all existing and available literature on the role of Ontology in Modeling knowledge in software development. Of the 150 initial papers located in well-known electronics journal databases. Forty-one papers were extracted through a multistage sifting process with independent validation of each step. These papers were assessed for the quality of the evidence they produced and further analyzed and categorized into the following 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 development are still immature and need a further empirical evaluation of practices in software development research

6.1 | Implications of The Study

This review has several implications for both researchers and practitioners. In terms of research, the review shows a need for more empirical studies that software developers 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 systematic review is the biased selection of the studies and the possible imprecision in data extraction from the variable sources. We implemented the following steps when developing our research strategy to eliminate this bias and ensure precision and accuracy in study selection. First, we treated the search string building process as a learning process that includes experimentation. We followed our research question to define keywords for extensive databases. Next, our study encompasses only articles that primarily 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 . In addition, we found that many articles lack sufficient information for inclusion in our study. More specifically, we found the level of detail at which the research method was described in 22 papers.

Furthermore, we are aware that for any of the papers describing challenges (challenges of ontology in software development), the underlying causes of the challenge didn't discuss in detail. It's hard to change the situation because we assumed that the authors of some studies might have chosen the challenges reported for specific reasons. However, the authors have focused on their specific research goals and questions, and the underlying causes for the challenges might have been overlooked.

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: Kristyanto M.A., Rishika R.V., Rawung F., Triandini E. (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*.