Accredited Ranking SINTA 2
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online on: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 7 No. 3 (2023) 564 - 578 ISSN Media Electronic: 2580-0760

Automatic Requirements Engineering: Activities, Methods, Tools, and Domains – A Systematic Literature Review

Rosa Delima[1], Khabib Mustofa[2], Anny Kartika Sari[3]
[1]Department of Informatics, Faculty of Information Technology, Universitas Kristen Duta Wacana
[2,3]Department of Computer Science & Electronics, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada
[1]rosadelima@staff.ukdw.ac.id*, [2]khabib@ugm.ac.id, [3]a_kartikasari@ugm.ac.id

Abstract

Requirements engineering (RE) is an initial activity in the software engineering process that involves many users. The involvement of various users in the RE process raises ambiguity and vagueness in requirements modeling. In addition, traditional RE is a time-consuming activity. Therefore various studies have been conducted to support process automation on RE. This paper conducts a systematic literature review (SLR) to obtain information about RE automation related to RE activities, methods/models, tools, and domains. SLR is done through 5 main stages: definition of research questions, conducting the search, screening for relevant papers, data extraction, mapping, and analysis. The data extraction and mapping are carried out on 155 relevant publications from 2016 to 2022. Based on the results from SLR, around 53% of the research focuses on RE automation in analysis and specifications, 40% focuses on elicitation, validation, and requirements management, and 7% focuses on requirements quality. NLP is the most used method in elicitation and specification, while for analysis, machine learning, NLP, and goal-oriented models are mostly used in automatic RE. Furthermore, many papers use specific models and methods for validation and requirements management. From the domain analysis results, it is obtained that more than half of the papers contribute directly to the RE domain, and some contribute to the development of RE automation in the software application domain.

Keywords: requirement engineering; automatic; systematic literature review; software engineering

1. Introduction

Requirements Engineering (RE) is an early step in the software development life cycle. This stage is very important to model user needs and ensure that the system developed can be useful and operated according to the objectives formulated. In the perspective of software engineering, RE is an activity that begins with communication between the system analysts and the stakeholders and continues with modeling system requirements activity [1]. RE includes a set of activities such as inception or feasibility study, elicitation, analysis, specification, verification/ validation, and requirements management [1], [2]. Inception or feasibility study is the initial stage of project preparation and analysis of the usefulness of the system to support business. Elicitation is a stage to collect needs carried out by various methods. Furthermore, the collected needs documents will be analyzed to formulate needs for users and for the system. Furthermore, the formulation of requirements will be modeled using the standard form of RE modeling. The model will be validated to determine the consistency and quality of RE. The final activity is to conduct management toward changes and traceability of RE activities [1], [2].

Interaction and communication with stakeholders is an important activities in RE. Stakeholders include the owner, management, and user. This activity can be difficult because stakeholders sometimes do not understand their needs toward the system to be developed, each stakeholder has different needs, and they can be opposite each other, and dynamic changes in the business and organizational environment [2]. The elicitation process involving stakeholders will produce a definition of needs in the form of a natural language. This has the potential to create ambiguous and vague in the needs' translation into the form of needs modeling in the specification stage [2]. Therefore, there have been many proposals for automation in various RE activities.

Automation is carried out at various stages at RE, starting from the elicitation stage to the management requirements. A systematic literature review (SLR) was carried out to map activities, techniques, and tools and identify the research that has been done to support the

Accepted: 10-02-2023 | Received in revised: 15-04-2023 | Published: 01-06-2023

automation of the RE process, a systematic literature review (SLR) was carried out. This paper is divided into some sections. Section 1 is the introduction, the background and related works related to systematic mapping in the RE field. Section 2 describes the applied methodology, followed by the results and analysis in Section 3, and the last part is formulated conclusions and future work.

In this section, the SLR that some researchers for the RE domain have carried out will be discussed, as well as the activities that will be carried out. Discussion of SLR-related studies was conducted to determine the scope of SLR topics that have been shown in the RE domain. The second part, it discusses the SLR, which includes the understanding and stages in implementing the SLR.

There have been some SLRs that are carried out related to RE, such as [3]- [13]. The RE-SLR related to the decision support system (DSS) was carried out by [3][4]. In his research [3], SLR was conducted to identify and classify DSS from a RE perspective. They identified 27 articles related to DSS in the RE process. The twenty-seven articles are classified into 39 models, 27 techniques, and 54 items of guidance. They found a gap in the literature on how to design a data warehouse and data flows in DSS [3]. Meanwhile, Wang et al. [4] conducted an SLR on 114 papers for the categorization of requirements traceability (RT) and evaluated 83 empirical studies related to decision support technology transfer for RT. Based on his research, they concluded ten major challenges in current RT activities, categorized existing RT techniques into six groups and 25 sub-groups, and identified seven potential future in RT [4].

SLR in RE was also done by [5][6] related to safety-critical systems (SCS). In their research [5], he reviewed 115 papers to find out the approach taken in the RE process for SCS started from elicitation activities to validation. Meanwhile, Vilela et al. [6] conducted an SLR to investigate the approach used to improve communication and integration between RE and safety engineering through the 57 papers reviewed.

RE and Agile methodology also received special attention for researchers. There were 3 SLRs carried out in this field, namely by [7][8][9]. In their research [7], he collected information on RE practices adopted and challenges faced by agile teams to understand how traditional requirements engineering issues are resolved using agile RE. Meanwhile, [8] focused on SLR to find out the state of the art of stakeholders and user involvement in Agile RE, and Curcio et al. [9] conducted SLRs to map RE in an agile context.

Other SLRs were carried out on RE in software product lines [10], real word settings [11], requirements change management [12], and requirements quality [13]. In their research, [10] conducted an SLR related to requirements modeling languages used in software product lines with respect to their degree of empirical validation, origin, the context of use, level of expressiveness, maturity, and industry adoption. Another SLR was carried out by [11] to get an overview of software patterns in real-world contexts during requirements engineering (RE) activities and conclude that patterns positively affect RE activities and can be used by system analysts to support their projects. The SLR related to requirements change management (RCM) was carried out by [12]; they reviewed 184 papers to gather information and analyze techniques used for RCM. The SLR regarding an automated approach to classifying quality requirements was carried out by Pérez-Verdejo et al., who conducted a review of 38 papers related to requirements and GitHub. Based on the research, there were seven quality attributes of the requirements: availability, fault tolerance, maintainability, performance, scalability, security, and usability [13].

The SLR summary in this related work can be seen in Table 1. Based on the data in Table 1, it is known that in the period 2015-2021, there have been 10 SLRs done on RE domains, and the SLRs have never been done for automatic RE. The SLR is important to be done because there has been a lot of research regarding the automation of the RE process and to find out the trends, methods, and processes of RE done by automation.

Table 1. Summary of SLR in RE

Literature Study	Number of Paper	SLR Topic and Description
[3]	27	DSS from an RE Perspective
[4]	114	Requirements traceability technologies and technology transfer decision support
[5]	115	Requirements engineering for safety-critical systems
[6]	57	Integration between requirements engineering and safety analysis
[7]	21	Agile requirements engineering practices and challenges
[8]	27	Agile Requirements Engineering with a focus on stakeholder and user involvement
[9]	104	Requirements engineering in agile software development
[10]	54	Requirements modeling languages for software product lines
[11]	22	Software patterns and requirements engineering activities in real-world settings
[12]	184	Techniques and practices used in requirements change management
[13]	38	An Automatic approach for quality requirements classification

The term systematic review is used to refer to specific research methodologies to obtain information and conduct evaluations in accordance with the focus of the study [14]. SLR is a systematic review conducted on evidence collected through literature. SLR has several

DOI: https://doi.org/10.29207/resti.v7i3.4924

advantages based on a well-defined methodology that minimizes bias in SLR results. This method can produce extensive and in-depth information based on empirical data from the results of research conducted, and for quantitative studies, this method allows for a combination of data using meta-analytic techniques [15].

SLR is carried out using a three-stage methodology that includes concepts, research, and results. The concept step involves formalizing the problem through a research question that will serve as the foundation for information extraction from the collected data. Studies are the stage of doing a review by analyzing the content, contrasting, or connecting components of the available evidence in the second stage. The compilation result will be built primarily on the study findings. The analysis and synthesis of the data will provide new data and knowledge that will be presented in conclusions at the findings stage. SLR has three basic procedures based on these three phases, including planning, review execution, and result analysis. [14]. At the SLR, planning tasks include building review methods, proposing research topics, and identifying needs. Activities for identification, selecting primary research, assessing the caliber of studies, data extraction, and monitoring are included in the execution phase of the review. Data synthesis and analysis are tasks associated with reviewing outcomes. The results are displayed in a review report. [15].

Figure 1 illustrates the processes of the systematic mapping study with reference to the three stages of the SLR for analysis of the classification and mapping process of the obtained data [16]. The SLR process entails five stages, including the formulation of research questions, conducting searches, screening papers, key wording using abstracts, and data extraction mapping process. The results of each stage are the review of the scope, all papers, relevant papers, classification scheme, and systematic map, respectively.

2. Research Methods

This section discusses the methodology used to conduct SLR. The activities carried out refer to figure 1.

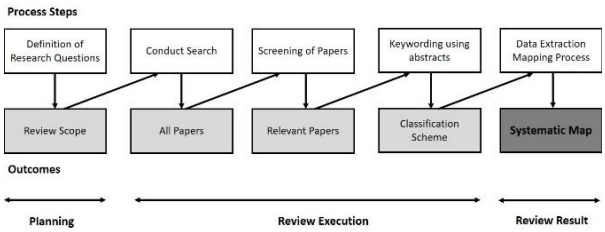

Figure 1. Systematic Literature Review and Mapping Diagram [15], [16].

Figure 1 begins with the identification of the need for SLR, followed by research question formulation, the searching process using keywords, screening of papers, keywording using abstract, and the data extraction mapping process.

2.1. Identification of The Need for SLR

RE is an important initial stage as the basis for designing and making programs in software engineering activities. In the RE, extracting information from stakeholders is done. Interaction and communication with stakeholders require a lot of energy and are time-consuming. Therefore automation in a series of RE activities is considered to have the opportunity to save time and energy in the RE process. SLR needs to be done to find out the extent to which the automation process has been carried out in the RE. This SLR aims to explore information and map processes, methods, and tools that researchers have carried out. Through this SLR, the next research opportunity will be formulated to support the process of automation on RE.

2.2. Definition of Research Questions

The purpose of this SLR is to learn more about automatic RE. The SLR is crucial since there has been much research into the automation of the RE process. This research wants to identify the trends, techniques, and procedures of RE carried out by automation. A research question (RQ), which would serve as the direction for the SLR analysis, was established to learn more about a more structured study. Based on the objectives of the SLR, the research questions (RQ) are formulated as follows:

RQ1: What are the steps in the RE that have been attempted for automation?

Rationale: RE has a series of activities namely elicitation, analysis, specification, validation and requirements management. Through this SLR, it will be known on what activities automation has been carried out, so that we can know the stages that have been studied and see opportunities for RE automation.

RQ2: What methods are used to support automation?

Rationale: The automation process needs to be supported by a series of methods. Through this RQ, a method mapping will be used to support automation in relation to the process on RE. Through mapping, we can find out the most commonly used methods and opportunities for implementing and developing methods for the automation process on RE.

RQ3: What are the tools that have been developed for automation process in RE?

Rationale: Knowing the tools that have been developed for RE automation provides opportunities for using tools and the possibility of developing new tools.

RQ4: What are the domains that have had RE automation efforts?

DOI: https://doi.org/10.29207/resti.v7i3.4924

Rationale: Through this RQ, fields or systems that are widely researched and targeted as case studies for the implementation of automatic RE will be known so that the opportunity for the application of automatic RE for future research will be revealed.

2.3. Conduct a Search based on keywords

This SLR's goal is to investigate information on RE automation processes. Thus, the search process's important keywords are:

("Requirement Engineering") and ("automatic" or "automated")

Four literature databases were searched, including Science Direct, Scopus, Proquest, and IEEE. The publishing years 2016 to 2022 and "journal articles" are the main search criteria.

2.4. Screening Paper for Inclusion and Exclusion.

There were 423 publications found after using particular keywords to target journal articles. The papers were then screened based on inclusion and exclusion criteria. Paper that was not pertinent to RQ was filtered using inclusion and exclusion criteria [15]. The identical material was filtered out using inclusion, and irrelevant paper was filtered out using exclusion.

Two steps of filtering were used for the exclusion factor, such as (1) abstract and keyword checking and (2) the writer's reading of the abstract. The word "requirement" was checked for in the abstract as part of the abstract checking procedure. After passing the initial exam, the writer conducted an abstract reading to determine the paper's applicability. According to the screening's findings, 155 papers were pertinent. Figure 2 shows the specifics of the filtered paper from each stage.

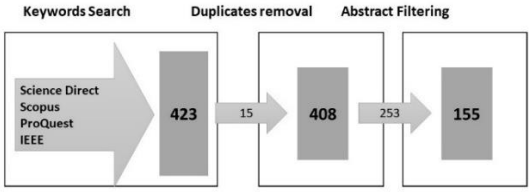

Figure 2. Screening Phases and Number of Selected Papers

2.5. Key Wording of Abstracts.

Using keywords can speed up the classification of SLRs. Keywording is done by reading abstracts and selecting keywords to create a classification system that is tailored to the study's environment. [16]. The classification scheme in Table 2 shows the important wording results.

Table 2. Classification Scheme

RQ	Classes / Categories
RQ1	Elicitation; Analysis; Specification; Validation/Verification; Requirement Management; RE Quality; RE Framework
RQ2	NLP; Goal Oriented Model; Ontology; Classification or Clustering Algorithm; Machine Learning; Logic or Formal Logic; Graph Model; IR or Semantic Web; Reuse Model; Rule-Based Model; Data/Text/Web Mining; Use Case Model; Model Based Approach; Specific Model; Specific Method; Specific Algorithm; Specific Technique/Tools
RQ3	Develop tool or prototype system; Using available support tools; Without Tool support
RQ4	RE Domain; Software Application Domain; Business Domain

3.6. Data Extraction and Mapping Studies.

Data extraction is based on 155 relevant papers according to the classification scheme that has been formulated. Excel table is used to facilitate the extraction and mapping process. Based on the final table, the number of frequencies for each category can be seen in Figure 3. The results of the mapping in Figure 3 focus on the number of papers related to 5 activities in RE, namely elicitation, analysis, specification, validation and requirements management.

3. Results and Discussions

Based on the established RQ, the SLR results and analysis are organized in this section. Based on the screening process, 155 relevant papers were distributed within 7 years of publication, from 2016 to 2022. This section will discuss the research findings and discussions based on the RQ.

3.1. RQ1: What are the steps in the RE that have been attempted for automation?

The RE process consists of several activities. In this SLR, activities are categorized into five such elicitation, analysis, specification, validation and requirements management (RM). Elicitation involves the activity of gathering information on the needs of stakeholders using various methods. The analysis includes the activities of classifying, determining priority, feature extraction, and other activities of analysis to formulate the needs of the user and system. The specification is the process of pouring analysis results into modeling requirements. The next important RE activity is to validate the modeling results. Validation includes checking validity, consistency, completeness, realism, and verifiability. The last activity of RE is requirements management which includes requirements identification, change management processes, traceability policies, and support tools [2]. The results of the SLR studies in figure 3 showed that from the 155 papers analyzed, only 142 papers are specifically supported 5 RE activities, while six papers examined the automation process for requirements quality (RQ), and seven papers proposed a specific framework for RE.

DOI: https://doi.org/10.29207/resti.v7i3.4924

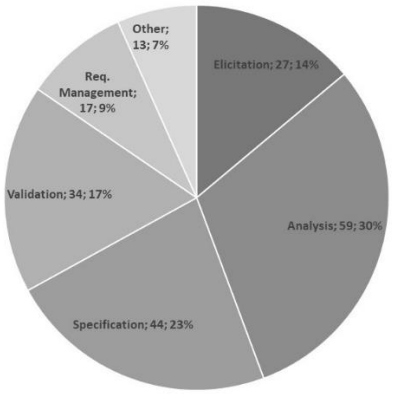

Figure 3. The number and percentage of paper for each RE activity.

Based on the 142 papers analyzed, it was found that the activities most supported by research for the automatic process were analysis followed by the specifications, each being as many as 59 and 44 papers. In comparison, research for the automation process of validation, elicitation, and RM has as many as 34, 27, and 17 papers (Figure 3).

The research also found that 27 papers automated more than one RE Activities. Based on research it is known that 22 papers (14%) automate 2 RE activities, while six papers (4%) automate three activities, and one paper (1%) automates 4 RE Activities.

The results of the analysis to answer RQ1 (Figure 3), it's known that around 53% of the research focuses on analysis activities and specifications of RE, 40% focuses on elicitation, validation, and requirements management activities, and the other 7% focuses on requirements quality and implementing or creating a framework for RE. This shows that the automation effort includes all stages in RE and proves that automation can be done on every RE activity. Some papers only discuss the automation of RE in only one process, only 28 papers examine automation in more than one process, and there has not been any research on the automation of overall RE activities. This condition opens the next research opportunity to be able to automate not only one RE activity.

3. 2. RQ2: What method is used to support automatic RE?

To answer RQ2, a review of the methods used by researchers to develop automatic RE was conducted. The result of the analysis shows that many methods/algorithms/models were used in the study. This method is categorized into 16 categories as Natural Language Processing (NLP), Goal Oriented Model, Ontology, Classification or Clustering Algorithm, Graph Model, Machine Learning, Logic/Formal Logic, Reuse Model, Rule-based model, Data/Text/Web Mining, Information Retrieval (IR) or Semantic Web, Use Case Model, Model-Based Approach, specific algorithm, specific method, and specific technique/tools. The number and percentage of papers for each category, along with the references, can be seen in Figures 4 and 5. Meanwhile, the percentage distribution of method implementation for each RE activity starting from elicitation to requirements management can be seen in Figures 6 to 10.

Figures 4 and 5 show that the three most widely used methods for automating the RE process are NLP with 52 papers or 20%, then machine learning and goal-oriented with 22 papers/9% and 20 papers (8%). Notably, 35 papers (14%) develop RE automation by applying a specific model. The list of specific models can be seen in Table 3.

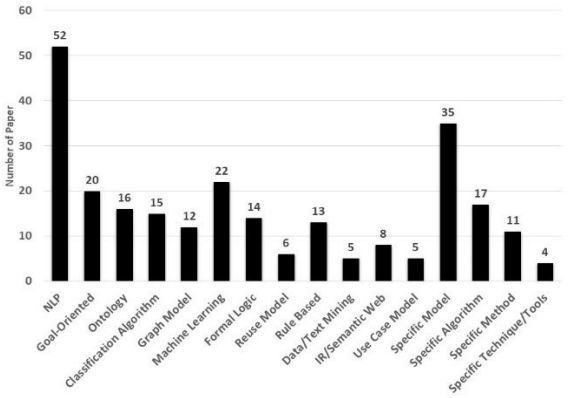

Figure 4. The number of papers based on the method used in RE.

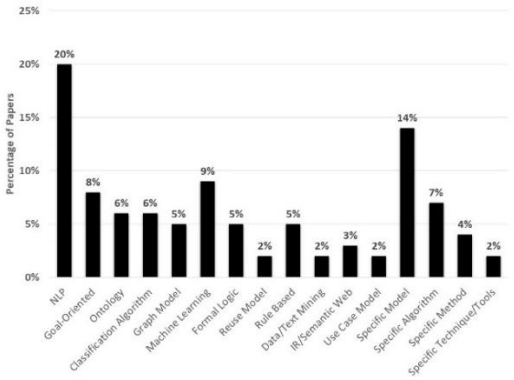

Figure 5. The percentage of papers based on the method used in RE.

For RE automation at each stage, it can be seen in Figure 6, that NLP, Machine Learning, Specific models, and goal-oriented are the four most widely used methods in automating elicitation stages in RE. Meanwhile, for the analysis stages in Figure 7, the NLP method, machine learning, Goal-Oriented, and formal approaches predominate for the automation of analysis activities in RE. The NLP method is dominant for the specifications stage in Figure 8, followed by rule-based and specific models.

The NLP approach is widely used in the three stages of RE. Various techniques are used in the NLP approach, which includes: POS tagging, tokenization, parsing, stop-words removal, term extraction, stemming,

DOI: https://doi.org/10.29207/resti.v7i3.4924

lemmatization, similarity measures, TF-IDF, sentence splitting, syntactic structure, linguistic rules, chunking, ML classification, NER, SRL, syntactic pattern, frequency analysis, semantic analysis, lexical analysis, keywords searching, text annotation, bag-of-word, regular expression, semantic annotation, clustering, lexical patterns, words searching, porter stemming, syntactic analysis, mapping rules, and VSM [17].

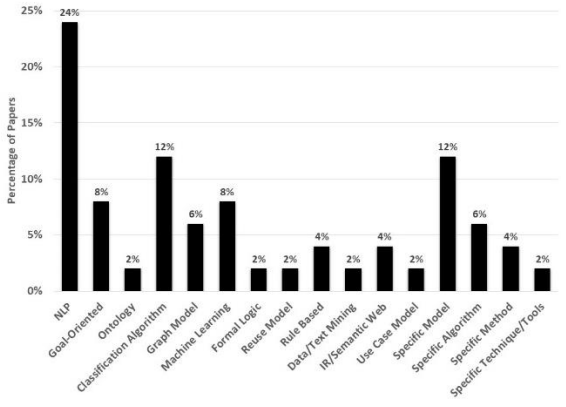

Figure 6. The percentage of papers based on the method used in elicitation.

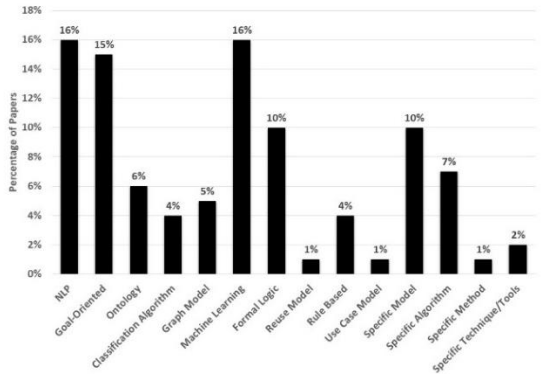

Figure 7. The percentage of papers based on the method used in analysis.

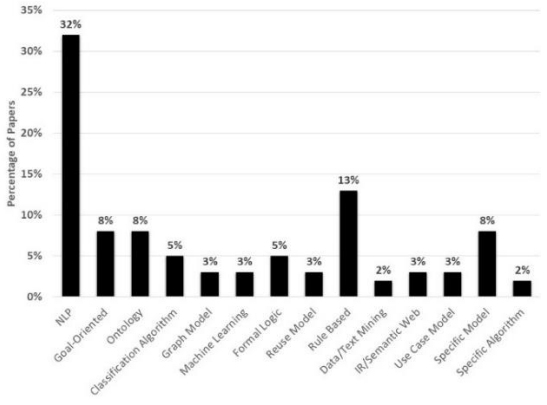

Figure 8. The percentage of papers based on the method used in specification.

Unlike the first three stages, in the validation process (Figure 9) and requirements management (Figure 10), the specific model approach is most widely used in the validation stage. Other methods that are quite widely used at this stage are Ontology, formal logic, and rule-based.

In Requirement management (Figure 10), the specific method dominates 25% for stage automation. The specific model, specific method, and specific algorithm approaches used in RE automation can be seen in Table 3.

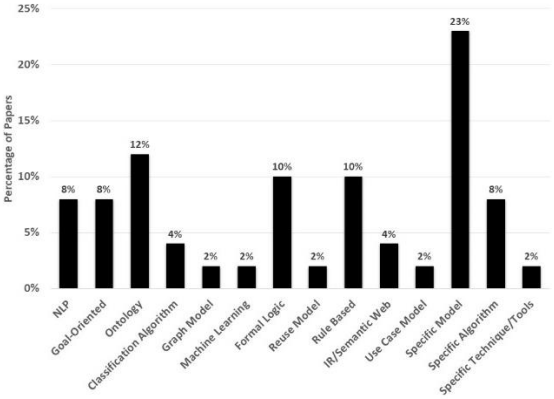

Figure 9. The percentage of papers based on the method used in validation.

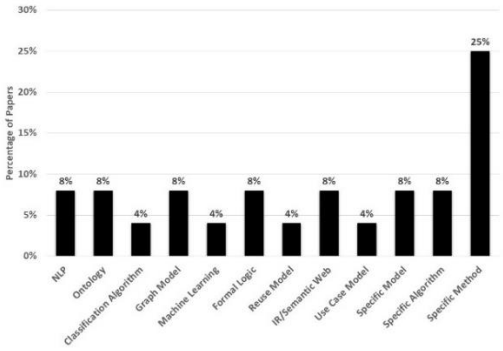

Figure 10. The percentage of papers based on the method used in requirement management.

Through analysis of the data, it is known that some papers do not only apply one method in their research. There were 22 papers applying two methods, 12 papers applying three methods, four papers applying four, and 1 paper applying five methods. The percentage of papers that applies more than one method can be seen in Figure 11.

Through analysis carried out to answer RQ2, it is known that a number of methods have been used to support automatic RE. There are still opportunities for in-depth exploration of methods that have been used or experimenting with new methods for RE automation. The combination of applying more than one method is often used by researchers. This is a fact that combining various methods is still an important issue to produce easier, more accurate, efficient, and effective automation.

DOI: https://doi.org/10.29207/resti.v7i3.4924

Table 3. Methods and references for specific models, methods, algorithms, and techniques/tools categories

Category	Literature and Method
Model-Based Approach	Feature model [18], [19]; Pattern-based model [20], [21]; MAPE-K model [22], [23]; Agent / multiagent-based model [24] - [28]; Viewpoint model [29] ,[30]; Problem-Based model [31], [32]; Continues Collaborative Model (CCM) [33]; Integrated-Secure SDLC model (IS-SDLC) [34]; Requirements frame model [35]; Grid Model [36]; Probabilistic model checking [37]; BIP (Behavior – Interaction – Priorities) Model [38]; [29], [39] model based testing; [40] model driven approach (MDA); [20] Model Driven Development (MDD); BERT Model [41] - [43]; RE-BERT Model [44]; Object-Role Modelling (ORM) notation [45]; i* model [46],[47]; scenario based modeling [48]; QR (Quality Requirements) Pattern [49] Annotated BPMN Model [50]
Specific Method	Speech recognition [51]; Statistic Method [52], [53]; Safety requirements specification (SRS) method [54]; Just in time (JIT) Requirements method [55]; Invariant Refinement Method for Self-Adaptation [56]; measurement-based method [57]; COSMIC Function Point Method [58]; Requirements change propagation method [59]; Requirements Description Schema [60]; Data-Driven Approach [49];
Specific Algorithm	Analytic Hierarchy Process (AHP) [61], [18]; DSS – multi-criteria [52]; Matching algorithm [62]; Quantum Algorithm [63]; Fuzzy Logic [64] [65]; PageRank Algorithm [41]; Decision Support [66], [67], [28]; Markov decision [68]; Genetic Algorithm [69], [70]; Horspool's Algorithm [71]; Key-phrase algorithm [72]; Naïve Bayes's Algorithm [73]
Specific technique/Tools	using tool Nomos and NomosT [74]; Automated Planning Technique [75]; Ethnographical [76]; DT-Golog [68];

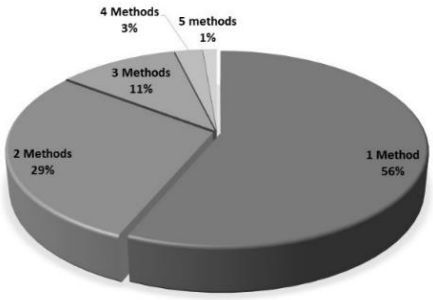

Figure 11. The Percentage of Papers that apply a combination of methods in RE automation

3.3 RQ3: What tools have been developed for the automation process in RE?

A review of the development and use of tools to support RE automation is done by mapping the process and results of the review paper. The mapping is done by categorizing the paper into three categories: automation supported by the tools developed, using tools that are already available, and without specifying tools, which means that the paper only discusses the framework or model without conducting trial and implementation. The review results show that 57 or 37% of papers use tools to support their research. While 32% or 50 papers only use the tools that are already available, and 31% of the papers do not define the development and use of tools in their research.

Analysis to answer RQ3 produces information in the form of new tools developed for RE automation and a number of tools available that can be used to support RE automation. The tools that have been developed and are available tools can be the main supporting factors in further research to maximize the automation process on RE.

3.4 RQ4: What are the domains that have been in efforts for RE automation?

A study of the papers' subject matter is done to address RQ4. Domains are categorized into three, such RE, software application, and business domain. There were 92 papers (59%) including the RE domain category, 53 papers (34%) discussing the development of RE automation in the software application domain, and 11 papers (7%) papers in the business company domain category. Particularly for the software application domain, it is categorized into seven software categories [1], such as software systems, application software, engineering/scientific software, embedded software, product line software, web/mobile applications, and artificial intelligence software. Business company domains include industrial, software engineering companies, small businesses, and start-ups company.

Based on this review, we obtain information that more than half of the papers contribute directly to the RE domain, and some contribute to the development of RE automation in the software application domain. Based on these data, it is known that direct RE automation has not been done much in the business environment so research to automate RE in the business environment needs to be explored to obtain facts about problems, obstacle, and solutions for RE automation directly.

3.5 Opportunity for Future Research.

The development of prospects for more study is done based on the findings and analysis of the SLR, and includes: Research to automate more than one RE activity can be essential in achieving comprehensive RE automation; Researchers have commonly done the use of specific models, methods, and algorithms, this indicates that there are still many possibilities for developing new models, methods, or algorithms for RE automation; Research combining more than one method is still an important issue to provide easier, more accurate, efficient and effective automation results; Research for the development of new tools, frameworks or models is also still an issue in RE automation; and The business company domain is a domain that still requires a lot of research to explore opportunities, obstacles, and models for RE automation.

DOI: https://doi.org/10.29207/resti.v7i3.4924

4. Conclusion

There has been a lot of SLR in the RE field to obtain in-depth information and knowledge about RE. This paper presents an SLR to get information about the level of automation process on RE. SLR is done through five main stages: defining research questions, searching, screening for relevant papers, data extraction, mapping, and analysis. The SLR is done by formulating four RQs which include activities on RE: automation, methods, tools, and domains. Then the keywords are determined, and as many as 432 papers that match the keywords have been gathered. Following the screening process, 155 pertinent publications from 2016 to 2022 were found. After extracting and mapping the data, the analysis and formulation of the results of the SLR were then carried out.

This SLR provides results such as information that automatic RE has been done on RE activities, including elicitation, analysis, specification, validation, and requirements management. Most papers discuss the application of automation to only one RE activity. This SLR produces a mapping of 16 categories of methods/models/algorithms used, three categories of tools, and three domains of RE automation. NLP is the most widely used method for elicitation and specification activities, while for RE analysis, the most paper uses machine learning, NLP, and goal-oriented model, Specific model is widely used for validation, and for requirements management, most papers use specific method. The results of domain analysis produce information that more than half of the papers contribute directly to the RE domain, and others contribute to the development of RE automation in the software application domain. The final result of the analysis of the data is the formulation of further research opportunities for RE automation. A summary of the SLRs carried out can be seen in Table 4.

In future work, the author will conduct further research in the form of extracting information related to the application and analysis of opportunities for integrating participatory methods on RE. Analysis will also be carried out regarding the opportunity to implement Crowdsourcing RE to support automatic RE. After that, the research will be continued with the development of models by integrating participatory methods with other methods for RE automation that involve users as the main actors in the RE process.

Table 4. SLR summary for automated requirements engineering

Reference	Elicitation	Analysis	Specification	Validation	Requirements Management	Other Activities	NLP	Goal Oriented Model	Ontology	Classification or Clustering Algorithm	Graph Model	Machine Learning	Formal Logic	Reuse model	Rule Based Model	Data/Text/Web Mining	IR or Semantic Web	Use Case Model	Model Based	Specific model	Specific Algorithm	Specific Method	Specific Technique/Tools	Number of activities	Number of methods
[18]		√						√			√									√	√			1	4
[19]						√														√				1	1
[20]						√													√	√				1	2
[21]	√	√																√				√		2	2
[22]						√										√				√				1	2
[23]						√							√							√				1	2
[24]		√							√											√				1	2
[25]				√																√				1	1
[26]				√				√												√				1	2
[27]			√																	√				1	1
[28]	√	√	√	√			√	√							√					√	√			4	5
[29]				√															√	√				1	2
[30]			√																	√				1	1
[31]			√																	√				1	1
[32]	√																			√				1	1
[33]		√																		√				1	1
[34]					√															√				1	1
[35]				√																√				1	1
[36]	√																			√				1	1
[37]		√	√	√				√												√				3	2
[38]				√																√				1	1
[39]				√									√						√					1	2
[40]			√				√												√					1	2
[41]					√							√								√	√			1	3
[42]		√																		√				1	1
[43]		√										√								√				1	2
[44]	√																			√				1	1
[45]		√																	√					1	1
[46]						√														√				1	1
[47]			√	√				√												√				2	2
[48]		√	√	√																√				3	1
[49]	√																					√	√	1	2
[50]				√																√				1	1
[51]	√															√						√		1	2
[52]						√	√				√					√					√	√		1	5

DOI: https://doi.org/10.29207/resti.v7i3.4924

Reference	Elicitation	Analysis	Specification	Validation	Requirements Management	Other Activities	NLP	Goal Oriented Model	Ontology	Classification or Clustering Algorithm	Graph Model	Machine Learning	Formal Logic	Reuse model	Rule Based Model	Data/Text/Web Mining	IR or Semantic Web	Use Case Model	Model Based	Specific model	Specific Algorithm	Specific Method	Specific Technique/Tools	Number of activities	Number of methods
[53]					√																	√		1	1
[54]					√																	√		1	1
[55]						√																√		1	1
[56]					√																	√		1	1
[57]					√																	√		1	1
[58]		√							√													√		1	2
[59]					√																	√		1	1
[60]					√																	√		1	1
[61]	√																				√			1	1
[62]			√				√			√	√										√			1	4
[63]		√																			√			1	1
[64]					√																√			1	1
[65]		√						√													√			1	2
[66]		√					√														√			1	2
[67]				√						√											√			1	2
[68]		√						√													√		√	1	3
[69]						√						√									√			1	2
[70]				√											√						√			1	2
[71]	√						√								√						√			1	3
[72]				√			√			√											√			1	3
[73]	√						√														√			1	2
[74]		√																					√	1	1
[75]	√											√											√	1	2
[76]				√																			√	1	1
[77]	√		√					√																2	1
[78]	√		√				√		√		√						√							2	4
[79]	√	√					√							√			√							2	3
[80]		√	√	√									√											3	1
[81]		√	√					√																2	1
[82]		√			√		√																	2	1
[83]		√	√	√				√																3	1
[84]			√	√			√	√							√									2	3
[85]			√	√			√																	2	1
[86]			√	√			√											√						2	2
[87]	√						√			√	√													1	3
[88]	√							√																1	1
[89]	√						√			√														1	2
[90]	√						√			√														1	2
[91]	√												√											1	1
[92]	√									√		√												1	2
[93]	√						√																	1	1
[94]	√							√																1	1
[95]	√						√			√		√												1	3
[96]		√										√												1	1
[97]		√										√												1	1
[98]		√										√												1	1
[99]		√					√			√														1	2
[100]		√										√			√									1	2
[101]		√					√																	1	1
[102]		√						√					√		√									1	3
[103]		√					√																	1	1
[104]		√						√																1	1
[105]		√						√	√				√											1	3
[106]		√					√					√		√										1	3
[107]		√											√											1	1
[108]		√						√																1	1
[109]			√						√															1	1
[110]			√				√																	1	1
[111]			√				√																	1	1
[112]			√				√								√									1	2
[113]			√				√							√			√							1	3
[114]			√															√						1	1
[115]			√				√					√												1	2
[116]			√						√	√						√								1	3
[117]			√												√									1	1
[118]			√											√										1	1
[119]				√			√																	1	1
[120]				√					√															1	1
[121]				√			√	√	√									√						1	4
[122]				√					√															1	1
[123]				√					√						√									1	2
[124]				√													√							1	1
[125]				√					√					√			√							1	3
[126]					√						√													1	1

DOI: https://doi.org/10.29207/resti.v7i3.4924

Reference	Elicitation	Analysis	Specification	Validation	Requirements Management	Other Activities	NLP	Goal Oriented Model	Ontology	Classification or Clustering Algorithm	Graph Model	Machine Learning	Formal Logic	Reuse model	Rule Based Model	Data/Text/Web Mining	IR or Semantic Web	Use Case Model	Model Based	Specific model	Specific Algorithm	Specific Method	Specific Technique/Tools	Number of activities	Number of methods
[127]					√	√	√			√	√													2	3
[128]					√		√							√			√							1	3
[129]					√				√								√							1	2
[130]					√								√					√						1	2
[131]					√				√															1	1
[132]					√								√											1	1
[133]						√	√																	1	1
[134]						√										√								1	1
[135]		√					√					√												1	2
[136]			√				√																	1	1
[137]			√				√																	1	1
[138]	√	√	√									√												3	1
[139]		√					√																	1	1
[140]		√										√												1	1
[141]		√									√													1	1
[142]		√										√												1	1
[143]		√		√									√											2	1
[144]	√	√									√													2	1
[145]			√	√					√						√									2	2
[146]			√				√										√							1	2
[147]						√						√												1	1
[148]		√					√					√												1	2
[149]				√								√												1	1
[150]		√		√									√											2	1
[151]		√	√				√								√									2	2
[152]		√	√					√					√											2	2
[153]	√						√			√														1	2
[154]		√								√														1	1
[155]		√	√							√														2	1
[156]		√								√														1	1
[157]		√	√						√															2	1
[158]		√	√					√	√															2	2
[159]		√					√																	1	1
[160]						√	√				√													1	2
[161]		√	√	√									√											3	1
[162]		√					√																	1	1
[163]		√					√					√												1	2
[164]	√						√																	1	1
[165]			√				√								√									1	2
[166]			√	√			√				√				√									2	3
[167]			√				√																	1	1
[168]			√				√																	1	1
[169]		√					√																	1	1
[170]		√										√												1	1
[171]		√	√	√									√											2	1
[172]		√									√	√												1	2
[173]			√				√																	1	1

Acknowledgment

Special thanks to Computer Science and Electronics Department Universitas Gadjah Mada and Faculty of Information Technology Duta Wacana Christian University for providing facilities and funding for publishing this article.

References

[1] R. S. Pressman and B. R. Maxim, *Software Engineering A Practitioner's Approach*, Eighth Edi. New York: Mc Graw Hill Education, 2015.

[2] I. Sommerville, *Software Engineering Ninth Edition*, Ninth Edit. United States of America: Addison Wesley, 2011.

[3] S. García, O. Romero, and R. Raventós, "DSS from an RE Perspective : A systematic mapping," *J. Syst. Softw.*, vol. 117, pp. 488–507, 2016, doi: 10.1016/j.jss.2016.03.046.

[4] B. Wang, R. Peng, Y. Li, H. Lai, and Z. Wang, "Requirements traceability technologies and technology transfer decision support : A systematic review," *J. Syst. Softw.*, vol. 146, pp. 59–79, 2018, doi: 10.1016/j.jss.2018.09.001.

[5] L. E. G. Martins and T. Gorschek, "Requirements engineering for safety-critical systems : A systematic literature review," *Inf. Softw. Technol.*, vol. 75, pp. 71–89, 2016, doi: 10.1016/j.infsof.2016.04.002.

[6] J. Vilela, J. Castro, L. E. G. Martins, and T. Gorschek, "Integration between requirements engineering and safety analysis : A systematic literature review," *J. Syst. Softw.*, vol. 125, pp. 68–92, 2017, doi: 10.1016/j.jss.2016.11.031.

[7] I. Inayat, S. S. Salim, S. Marczak, M. Daneva, and S. Shamshirband, "A systematic literature review on agile requirements engineering practices and challenges," *Comput. Human Behav.*, vol. 51, pp. 915–929, 2015, doi: 10.1016/j.chb.2014.10.046.

[8] E.-M. Schön, J. Thomaschewski, and M. J. Escalona, "Agile Requirements Engineering: A systematic literature review," *Comput. Stand. Interfaces*, vol. 49, pp. 79–91, 2017, doi: 10.1016/j.csi.2016.08.011.

[9] K. Curcio, T. Navarro, A. Malucelli, and S. Reinehr, "Requirements engineering: A systematic mapping study in agile software development," *J. Syst. Softw.*, vol. 139, pp. 32–50, 2018, doi: 10.1016/j.jss.2018.01.036.

DOI: https://doi.org/10.29207/resti.v7i3.4924

[10] S. Sepúlveda, A. Cravero, and C. Cachero, "Requirements modeling languages for software product lines : A systematic literature review R," *Inf. Softw. Technol.*, vol. 69, pp. 16–36, 2016, doi: 10.1016/j.infsof.2015.08.007.

[11] J. L. Barros-justo, F. B. V Benitti, and A. L. Cravero-leal, "Software patterns and requirements engineering activities in real-world settings : A systematic mapping study," *Comput. Stand. Interfaces*, vol. 58, pp. 23–42, 2018, doi: 10.1016/j.csi.2017.12.001.

[12] S. Jayatilleke and R. Lai, "A systematic review of requirements change management," *In*, vol. 93, pp. 163–185, 2018, doi: 10.1016/j.infsof.2017.09.004.

[13] J. M. Pérez-Verdejo, J. Sánchez-García, J. O. Ocharán-Hernández, E. Mezura-Montes, and K. Cortés-Verdín, "Requirements and GitHub Issues: An Automated Approach for Quality Requirements Classification," *Program. Comput. Softw.*, vol. 47, no. 8, pp. 704–721, 2021, doi: 10.1134/S0361768821080193.

[14] J. Biolchini, P. G. Mian, A. C. C. Natali, and G. Travassos, "Systematic Review in Software Engineering," Ri de Janeiro, 2005. doi: 10.1007/978-3-540-70621-2.

[15] B. Kitchenham, "Guidelines for performing Systematic Literature Reviews in Software Engineering," 2007. doi: 10.1145/1134285.1134500.

[16] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic Mapping Studies in Software Engineering," in *Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering*, 2007, vol. 80, no. 2, pp. 68–77. doi: 10.1142/S0218194007003112.

[17] L. Zhao *et al.*, "Natural Language Processing for Requirements Engineering," *ACM Comput. Surv.*, vol. 54, no. 3, pp. 1–41, 2021, doi: 10.1145/3444689.

[18] M. Noorian, E. Bagheri, and W. Du, "Toward automated quality-centric product line configuration using intentional variability," *Softw. Evol. Process*, vol. 29, no. 9, pp. 1–26, 2017, doi: 10.1002/smr.1870.

[19] M. Bashari, E. Bagheri, and W. Du, "Self-adaptation of service compositions through product line reconfiguration," *J. Syst. Softw.*, vol. 144, pp. 84–105, 2018, doi: 10.1016/j.jss.2018.05.069.

[20] H. Agh and R. Ramsin, "A pattern-based model-driven approach for situational method engineering," *Inf. Softw. Technol.*, vol. 78, pp. 95–120, 2016, doi: 10.1016/j.infsof.2016.05.010.

[21] A. Mokhtarian, A. Kampmann, B. Alrifaee, S. Kowalewski, B. Lampe, and L. Eckstein, "Agile Requirement Engineering for a Cloud System for Automated and Networked Vehicles," in *2nd International Workshop on Autonomous Systems Design (ASD 2020)*, 2020, no. August. doi: 10.4230/OASIcs.ASD.2020.4.

[22] E. Zavala, X. Franch, J. Marco, A. Knauss, and D. Damian, "SACRE : Supporting contextual requirements ' adaptation in modern self-adaptive systems in the presence of uncertainty at runtime," *Expert Syst. Appl.*, vol. 98, pp. 166–188, 2018, doi: 10.1016/j.eswa.2018.01.009.

[23] J. M. T. Portocarrero *et al.*, "RAMSES : A new reference architecture for self-adaptive middleware in Wireless Sensor Networks," *Ad Hoc Networks*, vol. 55, pp. 3–27, 2017, doi: 10.1016/j.adhoc.2016.11.004.

[24] A. A. Lopez-lorca, G. Beydoun, R. Valencia-garcia, and R. Martinez-bejar, "Supporting agent oriented requirement analysis with ontologies," *J. Hum. Comput. Stud.*, vol. 87, pp. 20–37, 2016, doi: 10.1016/j.ijhcs.2015.10.007.

[25] Y. Abushark, J. Thangarajah, J. Harland, and T. Miller, "A Framework for automatically ensuring the conformance of agent designs," *J. Syst. Softw.*, vol. 131, pp. 266–310, 2017, doi: 10.1016/j.jss.2017.05.098.

[26] A. Douglas, T. Mazzuchi, and S. Sarkani, "A stakeholder framework for evaluating the -ilities of autonomous behaviors in complex adaptive systems," *Syst. Eng.*, vol. 23, no. 5, pp. 633–655, 2020, doi: 10.1002/sys.21555.

[27] E. A. Oliveira, V. Maram, and L. Sterling, "Transitioning from motivational goal models to user stories within user-centred software design," in *CEUR Workshop Proceedings*, 2022, vol. 3107.

[28] R. Delima, R. Wardoyo, and K. Mustofa, "Automatic Requirements Engineering Model using Goal-Oriented Modelling with Text Pre-Processing Technique," in *Sixth International Conference on Informatics and Computing (ICIC)*, 2021, pp. 1–8. doi: 10.1109/icic54025.2021.9632980.

[29] B. K. Aichernig, K. Hörmaier, F. Lorber, D. Nickovic, and S. Tiran, "Require , test , and trace IT," in *International Journal Software Tools Technology Transfer*, 2017, pp. 409–426. doi: 10.1007/s10009-016-0444-z.

[30] C. Khatwani, X. Jin, N. Niu, A. Koshoffer, L. Newman, and J. Savolainen, "Advancing viewpoint merging in requirements engineering : a theoretical replication and explanatory study," *Requir. Eng.*, vol. 22, no. 3, pp. 317–338, 2017, doi: 10.1007/s00766-017-0271-0.

[31] R. G. M. Souza and P. C. Stadzisz, "PROBLEM-BASED SOFTWARE REQUIREMENTS SPECIFICATION ESPECIFICAÇÃO DE REQUISITOS DE SOFTWARE," *Rev. Eletrônica Sist. Informação*, vol. 16 No. 2, pp. 1–26, 2016, doi: 10.21529/RESI.2016.1502002.

[32] R. Meis and M. Heisel, "Computer-Aided Identification and Validation of Privacy Requirements," *Information*, vol. 7, no. 2, p. 28, 2016, doi: 10.3390/info7020028.

[33] P. Cheng, J. Fu, and L. Chen, "Knowledge Transfer of Software Tool Development for Functional Requirements Analysis," *Comput. Appl. Enginerring Educ.*, vol. 24, no. 1, pp. 131–143, 2016, doi: 10.1002/cae.21679.

[34] M. Ramachandran, "Software security requirements management as an emerging cloud computing service," *Int. J. Inf. Manage.*, vol. 36, no. 4, pp. 580–590, 2016, doi: 10.1016/j.ijinfomgt.2016.03.008.

[35] Y. Matsumoto, S. Shirai, and A. Ohnishi, "A Method for Verifying Non-Functional Requirements," *Procedia Comput. Sci.*, vol. 112, pp. 157–166, 2017, doi: 10.1016/j.procs.2017.08.006.

[36] S. Dey and S. Lee, "REASSURE : Requirements elicitation for adaptive socio-technical systems using repertory grid," *Inf. Comput. Secur.*, vol. 87, pp. 160–179, 2017, doi: 10.1016/j.infsof.2017.03.004.

[37] D. F. Mendonca, G. N. Rodrigues, R. Ali, V. Alves, and L. Baresi, "GODA : A goal-oriented requirements engineering framework for runtime dependability analysis," *Inf. Softw. Technol.*, vol. 80, pp. 245–264, 2016, doi: 10.1016/j.infsof.2016.09.005.

[38] E. Stachtiari, A. Mavridou, P. Katsaros, S. Bliudze, and J. Sifakis, "Early validation of system requirements and design through correctness," *J. Syst. Softw.*, vol. 145, pp. 52–78, 2018, doi: 10.1016/j.jss.2018.07.053.

[39] J. Barnat, P. Bauch, N. Benes, L. Brim, J. Beran, and T. Kratochvila, "Analysing sanity of requirements for avionics," *Form. Asp. Comput.*, vol. 28, pp. 45–63, 2016, doi: 10.1007/s00165-015-0348-9.

[40] M. Elallaoui, K. Nafil, and R. Touahni, "Automatic Transformation of User Stories into UML Use Case Diagrams using NLP Techniques," *Procedia Comput. Sci.*, vol. 130, pp. 42–49, 2018, doi: 10.1016/j.procs.2018.04.010.

[41] I. Gräßler, C. Oleff, and D. Preuß, "Proactive Management of Requirement Changes in the Development of Complex Technical Systems," *Appl. Sci.*, vol. 12, 2022, doi: 10.3390/app12041874.

[42] G. Li, C. Zheng, M. Li, and H. Wang, "Automatic Requirements Classification Based on Graph Attention Network," *IEEE Access*, vol. 10, pp. 30080–30090, 2022, doi: 10.1109/ACCESS.2022.3159238.

[43] K. Bhatia and A. Sharma, "Sector classification for crowd-based software requirements," 2021. doi: 10.1145/3412841.3442005.

[44] A. F. De Araújo and R. M. Marcacini, "RE-BERT: Automatic extraction of software requirements from app reviews using BERT language model," 2021. doi:

DOI: https://doi.org/10.29207/resti.v7i3.4924

10.1145/3412841.3442006.
[45] K. Großer, V. Riediger, and J. Jürjens, "Requirements document relations: A reuse perspective on traceability through standards," *Softw. Syst. Model.*, 2022, doi: 10.1007/s10270-021-00958-y.
[46] Y. Wang, T. Li, Q. Zhou, and J. Du, "Toward practical adoption of i* framework: an automatic two-level layout approach," *Requir. Eng.*, vol. 26, no. 3, pp. 301–323, 2021, doi: 10.1007/s00766-021-00346-4.
[47] Y. Yang, Y. Bok, Z. Yang, E. Sheriff, and T. Li, "Goal2UCM: Automatic Generation of Use Case Model from iStar Model," in *Proceedings of the 14th International iStar Workshop*, 2021, pp. 21–27.
[48] C. Wiecher and J. Greenyer, "BeSoS: A Tool for Behavior-driven and Scenario-based Requirements Modeling for Systems of Systems," in *CEUR Workshop Proceedings*, 2021, vol. 2857.
[49] M. Oriol *et al.*, "Data-driven and tool-supported elicitation of quality requirements in agile companies," *Softw. Qual. J.*, vol. 28, pp. 931–963, 2020, doi: 10.1007/s11219-020-09509-y.
[50] Q. Ramadan, D. Strüber, M. Salnitri, J. Jürjens, V. Riediger, and S. Staab, "A semi-automated BPMN-based framework for detecting conflicts between security , data-minimization , and fairness requirements," *Softw. Syst. Model.*, vol. 19, pp. 1191–1227, 2020, doi: 10.1007/s10270-020-00781-x.
[51] S. Reddivari, T. Bhowmik, and C. Hollis, "Automated support to capture verbal just-in-time requirements via audio mining and cluster-based visualization," *J. Ind. Inf. Integr.*, 2018, doi: 10.1016/j.jii.2018.06.001.
[52] R. Pinguie, P. Véron, F. Segonds, and N. Croué, "A requirement mining framework to support complex sub-systems suppliers," *Procedia CIRP*, vol. 70, pp. 410–415, 2018, doi: 10.1016/j.procir.2018.03.228.
[53] H. Xie, J. Yang, C. K. Chang, and L. Liu, "A statistical analysis approach to predict user's changing requirements for software service evolution," *J. Syst. Softw.*, vol. 132, pp. 147–164, 2017, doi: 10.1016/j.jss.2017.06.071.
[54] K. Beckers, I. Côté, T. Frese, D. Hatebur, and M. Heisel, "A structured and systematic model-based development method for automotive systems, considering the OEM/supplier interface," *Reliab. Eng. Syst. Saf.*, vol. 158, pp. 172–184, 2017, doi: 10.1016/j.ress.2016.08.018.
[55] T. Bhowmik and A. Q. Do, "Refinement and resolution of just-in-time requirements in open source software and a closer look into non-functional requirements ☆," *J. Ind. Inf. Integr.*, 2018, doi: 10.1016/j.jii.2018.03.001.
[56] I. Gerostathopoulos *et al.*, "Self-adaptation in software-intensive cyber – physical systems : From system goals to architecture configurations," *J. Syst. Softw.*, vol. 122, pp. 378–397, 2016, doi: 10.1016/j.jss.2016.02.028.
[57] V. Antinyan and M. Staron, "Rendex : A method for automated reviews of textual requirements," *J. Syst. Softw.*, vol. 131, pp. 63–77, 2017, doi: 10.1016/j.jss.2017.05.079.
[58] S. Bagriyanik and A. Karahoca, "Automated COSMIC Function Point measurement using a requirements engineering ontology," *Inf. Softw. Technol.*, vol. 72, pp. 189–203, 2016, doi: 10.1016/j.infsof.2015.12.011.
[59] P. H. Hein, H. Nathaniel, and B. Markos, "Predicting requirement change propagation through investigation of physical and functional domains," *Res. Eng. Des.*, vol. 29, no. 2, pp. 309–328, 2018, doi: 10.1007/s00163-017-0271-6.
[60] T. Shah and S. V Patel, "A Novel Approach for Specifying Functional and Non-Functional Requirements using RDS (Requirement Description Schema)," *Procedia - Procedia Comput. Sci.*, vol. 79, pp. 852–860, 2016, doi: 10.1016/j.procs.2016.03.083.
[61] N. Argyropoulos, K. Angelopoulos, H. Mouratidis, and A. Fish, "Risk-aware decision support with constrained goal models," *Inf. Comput. Secur.*, vol. 26, no. 4, pp. 472–490, 2018, doi: 10.1108/ICS-01-2018-0010.
[62] D. Ko, S. Kim, and S. Park, "Automatic recommendation to omitted steps in use case specification," *Requir. Eng.*, pp. 1–28, 2018, doi: 10.1007/s00766-018-0288-z.
[63] A. C. Kumari and K. Srinivas, "Comparing the performance of quantum-inspired evolutionary algorithms for the solution of software requirements selection problem," *Inf. Softw. Technol.*, vol. 76, pp. 31–64, 2016, doi: 10.1016/j.infsof.2016.04.010.
[64] C. Hettiarachchi, H. Do, and B. Choi, "Risk-based test case prioritization using a fuzzy expert system," *Inf. Softw. Technol.*, vol. 69, pp. 1–15, 2016, doi: 10.1016/j.infsof.2015.08.008.
[65] C. W. Mohammad, M. Shahid, and S. Z. Hussain, "Fuzzy attributed goal oriented software requirements analysis with multiple stakeholders," *Int. J. Inf. Technol.*, vol. 13, no. 6, pp. 2345–2353, 2021, doi: 10.1007/s41870-017-0073-0.
[66] D. Rajpathak, P. M. Peranandam, and S. Ramesh, "Automatic development of requirement linking matrix based on semantic similarity for robust software development," *J. Syst. Softw.*, vol. 186, 2022, doi: 10.1016/j.jss.2021.111211.
[67] I. Hajri, A. Goknil, F. Pastore, and L. C. Briand, "Automating system test case classification and prioritization for use case-driven testing in product lines," *Empir. Softw. Eng.*, vol. 25, no. 5, pp. 3711–3769, 2020, doi: 10.1007/s10664-020-09853-4.
[68] S. Liaskos, S. M. Khan, and J. Mylopoulos, "Modeling and reasoning about uncertainty in goal models: a decision-theoretic approach," *Softw. Syst. Model.*, 2022, doi: 10.1007/s10270-021-00968-w.
[69] D. Adanza Dopazo, V. Moreno Pelayo, and G. Génova Fuster, "An automatic methodology for the quality enhancement of requirements using genetic algorithms," *Inf. Softw. Technol.*, vol. 140, no. July, 2021, doi: 10.1016/j.infsof.2021.106696.
[70] M. Aldekhail and M. Almasri, "Intelligent identification and resolution of software requirement conflicts: Assessment and evaluation," *Comput. Syst. Sci. Eng.*, vol. 40, no. 2, pp. 469–489, 2021, doi: 10.32604/CSSE.2022.018269.
[71] S. Alzahari and M. Kamalrudin, "An Approach to Elicit Trustworthiness Requirements in Blockchain technology," in *Journal of Physics: Conference Series*, 2021, vol. 1807. doi: 10.1088/1742-6596/1807/1/012031.
[72] M. Singh and G. S. Walia, "Automated mapping of fault logs to SRS requirements using key-phrase extraction," in *Proceedings of the 2021 ACMSE Conference - ACMSE 2021: The Annual ACM Southeast Conference*, 2021, pp. 138–145. doi: 10.1145/3409334.3452043.
[73] N. Kengphanphanit and P. Muenchaisri, "Automatic Requirements Elicitation from Social Media (ARESM)," in *CCCIS 2020*, 2020, pp. 57–62.
[74] N. Zeni, E. A. Seid, P. Engiel, and J. Mylopoulos, "NómosT : Building large models of law with a tool-supported process," *Data Knowl. Eng.*, vol. 117, pp. 407–418, 2018, doi: 10.1016/j.datak.2018.04.009.
[75] C. Liu, "An intelligent planning technique-based software requirement analysis," *Int. J. Comput. Sci. Eng.*, vol. 13, no. 3, pp. 285–295, 2016.
[76] Z. S. Li, C. Werner, N. Ernst, and D. Damian, "Towards privacy compliance: A design science study in a small organization," *Inf. Softw. Technol.*, vol. 146, no. February, 2022, doi: 10.1016/j.infsof.2022.106868.
[77] C. Kalloniatis, "Incorporating privacy in the design of cloud-based systems : a conceptual meta-model," *Inf. Comput. Secur.*, vol. 25 No.5, pp. 614–633, 2017, doi: 10.1108/ICS-06-2016-0044.
[78] E. Casagrande, E. Arnautovic, W. L. Woon, H. H. Zeineldin, and S. Davor, "Semiautomatic System Domain Data Analysis : A Smart Grid Feasibility Case Study," *IEEE Transcation Syst. Man, Cybernatics Syst.*, vol. 47, no. 12, pp. 3117–3127, 2017, doi: 10.1109/TSMC.2016.2562501.
[79] N. H. Bakar, Z. M. Kasirun, N. Salleh, and H. A. Jalab, "Extracting features from online software reviews to aid requirements reuse," *Appl. Soft Comput. J.*, vol. 49, pp. 1297–1315, 2016, doi: 10.1016/j.asoc.2016.07.048.
[80] S. Feo-arenis, B. Westphal, D. Dietsch, M. Muniz, S. Andisha, and A. Podelski, "Ready for testing : ensuring conformance to industrial standards through formal verification," *Form. Asp.*

DOI: https://doi.org/10.29207/resti.v7i3.4924

Comput., vol. 28, pp. 499–527, 2016, doi: 10.1007/s00165-016-0365-3.
[81] C. M. Nguyen, R. Sebastiani, P. Giorgini, and J. Mylopoulos, "Multi-objective reasoning with constrained goal models," *Requir. Eng.*, vol. 23, no. 2, pp. 189–225, 2018, doi: 10.1007/s00766-016-0263-5.
[82] J. Misra, "Terminological inconsistency analysis of natural language requirements," *Inf. Softw. Technol.*, vol. 74, pp. 183–193, 2016, doi: 10.1016/j.infsof.2015.11.006.
[83] M. Gharib and P. Giorgini, "Information quality requirements engineering with STS-IQ," *Inf. Softw. Technol.*, 2018, doi: 10.1016/j.infsof.2018.11.002.
[84] F. Bargui, H. Ben-abdallah, and J. Feki, "A natural language-based approach for a semi-automatic data mart design and ETL generation," *J. Decis. Syst.*, vol. 25, no. 4, pp. 1–36, 2016, doi: 10.1080/12460125.2016.1158066.
[85] A. C. C. Souza, F. L. S. Nunes, and M. E. Delamaro, "An automated functional testing approach for virtual reality applications," *Softw. Test. Verif. Realiability*, vol. 28, no. 8, pp. 1–31, 2018, doi: 10.1002/stvr.1690.
[86] P. X. Mai, A. Goknil, L. K. Shar, F. Pastore, L. C. Briand, and S. Shaame, "Modeling Security and Privacy Requirements : a Use Case-Driven Approach," *Inf. Softw. Technol.*, vol. 100, pp. 165–182, 2018, doi: 10.1016/j.infsof.2018.04.007.
[87] N. Niu *et al.*, "Requirements Socio-Technical Graphs for Managing Practitioners ' Traceability Questions," *IEEE Transections Comput. Soc. Syst.*, vol. 5, no. 4, pp. 1152–1162, 2018, doi: 10.1109/TCSS.2018.2872059.
[88] S. Woldeamlak, A. Diabat, and D. Svetinovic, "Goal-Oriented Requirements Engineering for Research-Intensive Complex Systems : A Case Study," *Syst. Eng.*, vol. 19, no. 4, pp. 322–333, 2016, doi: 10.1002/sys.
[89] W. Maalej, Z. Kurtanovic, H. Nabil, and C. Stanik, "On the automatic classification of app reviews," *Requir. Eng.*, vol. 21, pp. 311–331, 2016, doi: 10.1007/s00766-016-0251-9.
[90] I. Morales-ramirez, F. M. Kifetew, and A. Perini, "Speech-acts based analysis for requirements discovery from online discussions," *Inf. Syst.*, pp. 1–17, 2018, doi: 10.1016/j.is.2018.08.003.
[91] C. Neureiter, G. Eibl, D. Engel, S. Schlegel, and M. Uslar, "A concept for engineering smart grid security requirements based on SGAM models," *Comput. Sci. - Res. Dev.*, vol. 31, no. 1–2, pp. 65–71, 2016, doi: 10.1007/s00450-014-0288-2.
[92] E. Guzman, R. Alkadhi, and N. Seyff, "An exploratory study of Twitter messages about software applications," *Requir. Eng.*, vol. 22, no. 3, pp. 387–412, 2017, doi: 10.1007/s00766-017-0274-x.
[93] B. Aysolmaz, H. Leopold, H. A. Reijers, and O. Demirörs, "A semi-automated approach for generating natural language requirements documents based on business process models," *Inf. Softw. Technol.*, vol. 93, pp. 14–29, 2018, doi: 10.1016/j.infsof.2017.08.009.
[94] J. Horkoff, N. A. Maiden, and D. Asboth, "Creative goal modeling for innovative requirements," *Inf. Softw. Technol.*, pp. 1–16, 2018, doi: 10.1016/j.infsof.2018.09.005.
[95] C. Li, L. Huang, J. Ge, B. Luo, and V. Ng, "Automatically classifying user requests in crowdsourcing requirements engineering," *J. Syst. Softw.*, vol. 138, pp. 108–123, 2018, doi: 10.1016/j.jss.2017.12.028.
[96] A. Gupta and C. Gupta, "CDBR : A semi-automated collaborative execute-before-after dependency-based requirement prioritization approach," *J. King Saud Univ. - Comput. Inf. Sci.*, 2018, doi: 10.1016/j.jksuci.2018.10.004.
[97] F. Shao, R. Peng, H. Lai, and B. Wang, "DRank : A semi-automated requirements prioritization method based on preferences and dependencies," *J. Syst. Softw.*, vol. 126, pp. 141–156, 2017, doi: 10.1016/j.jss.2016.09.043.
[98] T. do N. Ferreira, A. A. Araújo, A. D. B. Neto, and J. T. De Souza, "Incorporating user preferences in ant colony optimization for the next release problem," *Appl. Soft Comput. J.*, vol. 49, pp. 1283–1296, 2016, doi: 10.1016/j.asoc.2016.06.027.
[99] A. Rago, C. Marcos, and J. A. Diaz-pace, "Using semantic roles to improve text classification in the requirements domain," *Lang. Resour. Eval.*, vol. 52, no. 3, pp. 801–837, 2018, doi: 10.1007/s10579-017-9406-7.
[100] Q. Khan, M. A. Khan, Q. Javaid, I. Ullah, K. Ullah, and M. Fawad, "A Rule Based Genetic Algorithm Technique for Conflicts Resolution in Requirements Engineering," vol. 13, no. 11, pp. 8427–8433, 2016, doi: 10.1166/jctn.2016.5993.
[101] A. Mustafa, W. M. N. W. Kadir, and N. Ibrahim, "Automated Natural Language Requirements Analysis using General Architecture for Text Engineering (GATE) Framework," *J. Telecommun. Electron. Comput. Eng.*, vol. 9, no. 3, pp. 97–101, 2018.
[102] R. Rios, C. Fernandez-gago, and J. Lopez, "Modelling privacy-aware trust negotiations," *Comput. Secur.*, vol. 77, pp. 773–789, 2018, doi: 10.1016/j.cose.2017.09.015.
[103] G. Lucassen, M. Robeer, F. Dalpiaz, J. M. E. M. van der Werf, and S. Brinkkemper, "Extracting conceptual models from user stories with Visual Narrator," *Requir. Eng.*, vol. 22, no. 3, pp. 339–358, 2017, doi: 10.1007/s00766-017-0270-1.
[104] J. Horkoff and E. Yu, "Interactive goal model analysis for early requirements engineering," *Requir. Eng.*, vol. 21, pp. 29–61, 2016, doi: 10.1007/s00766-014-0209-8.
[105] C. Diamantini, A. Freddi, S. Longhi, D. Potena, and E. Storti, "A goal-oriented , ontology-based methodology to support the design of AAL environments," *Expert Syst. Appl.*, vol. 64, pp. 117–131, 2016, doi: 10.1016/j.eswa.2016.07.032.
[106] W. Xiong, Z. Lu, B. Li, B. Hang, and Z. Wu, "Automating smart recommendation from natural language API descriptions via representation learning," *Futur. Gener. Comput. Syst.*, vol. 87, pp. 382–391, 2018, doi: 10.1016/j.future.2018.05.006.
[107] N. Daclin, S. M. Daclin, V. Chapurlat, and B. Vallespir, "Writing and verifying interoperability requirements : Application to collaborative processes," *Comput. Ind.*, vol. 82, pp. 1–18, 2016, doi: 10.1016/j.compind.2016.04.001.
[108] H. Kaiya and K. Haga, "A CASE tool for Goal Dependency Model with Attributes based on An Existing UML Editor," *Procedia Comput. Sci.*, vol. 112, pp. 1196–1205, 2017, doi: 10.1016/j.procs.2017.08.033.
[109] F. Gailly, N. Alkhaldi, S. Casteleyn, and W. Verbeke, "Recommendation-Based Conceptual Modeling and Ontology Evolution Framework (CMOE +)," *Bus. Inf. Syst. Eng.*, vol. 59, no. 4, pp. 235–250, 2017, doi: 10.1007/s12599-017-0488-y.
[110] I. Hajri, A. Goknil, L. C. Briand, and T. Stephany, "Configuring use case models in product families," *Softw. Syst. Model.*, vol. 17, no. 3, pp. 939–971, 2018, doi: 10.1007/s10270-016-0539-8.
[111] A. Rago, C. Marcos, and J. A. Diaz-pace, "Identifying duplicate functionality in textual use cases by aligning semantic actions," *Softw. Syst. Model.*, vol. 15, pp. 579–603, 2016, doi: 10.1007/s10270-014-0431-3.
[112] M. Z. Alksasbeh, B. A. Y. Alqaralleh, T. A. Alramadin, and K. A. Alemarien, "AN AUTOMATED USE CASE DIAGRAMS GENERATOR," *J. Theor. Appl. Infromation Technol.*, vol. 95, no. 5, pp. 1182–1190, 2017.
[113] J. M. C. De Gea, J. Nicolás, J. L. Fernández-alemán, and T. Ambrosio, "Automated support for reuse-based requirements engineering in global software engineering," *Softw. Evol. Process*, vol. 29, no. 8 e1873, pp. 1–16, 2017, doi: 10.1002/smr.1873.
[114] M. Zhang *et al.*, "Specifying uncertainty in use case models," *J. Syst. Softw.*, vol. 144, pp. 573–603, 2018, doi: 10.1016/j.jss.2018.06.075.
[115] A. Al-hroob, A. T. Imam, and R. Al-heisa, "The use of artificial neural networks for extracting actions and actors from requirements document," *Inf. Softw. Technol.*, vol. 101, pp. 1–15, 2018, doi: 10.1016/j.infsof.2018.04.010.
[116] J. Guo, M. Gibiec, and J. Cleland-Huang, "Tackling the term-mismatch problem in automated trace retrieval," *Empir. Softw. Eng.*, vol. 22, no. 3, pp. 1103–1142, 2017, doi: 10.1007/s10664-016-9479-8.

DOI: https://doi.org/10.29207/resti.v7i3.4924

[117] A. Benabbou, S. N. Bahloul, and P. Dhaussy, "An Automated Transformation Approach for Requirement Specification," *Procedia - Procedia Comput. Sci. - Inf. Technol. Quant. Manag.*, vol. 91, pp. 891–900, 2016, doi: 10.1016/j.procs.2016.07.107.

[118] S. Ouhbi, J. L. Fernández-alemán, J. M. Carrillo-de-gea, A. Toval, and A. Idri, "E-health internationalization requirements for audit purposes," *Comput. Methods Programs Biomed.*, vol. 144, pp. 49–60, 2017, doi: 10.1016/j.cmpb.2017.03.014.

[119] M. Kamalrudin, J. Hosking, and J. Grundy, *MaramaAIC : tool support for consistency management and validation of requirements*, vol. 24, no. 1. Springer US, 2017. doi: 10.1007/s10515-016-0192-z.

[120] R. Sinha, C. Pang, G. S. Martinez, and V. Vyatkin, "Automatic test case generation from requirements for industrial cyber-physical systems," *Automatisierungstechnik*, vol. 64, no. 3, pp. 216–230, 2016, doi: 10.1515/auto-2015-0075.

[121] T. H. Nguyen, J. C. Grundy, and M. Almorsy, "Ontology-based automated support for goal – use case model analysis," *Softw. Qual. J.*, vol. 24, no. 3, pp. 635–673, 2016, doi: 10.1007/s11219-015-9281-7.

[122] B. Tenbergen, T. Weyer, and K. Pohl, "Hazard Relation Diagrams : a diagrammatic representation to increase validation objectivity of requirements-based hazard mitigations," *Requir. Eng.*, vol. 23, no. 2, pp. 291–329, 2018, doi: 10.1007/s00766-017-0267-9.

[123] C. Liu, "CDNFRE : Conflict detector in non-functional requirement evolution based on ontologies," *Comput. Stand. Interfaces*, vol. 47, pp. 62–76, 2016, doi: 10.1016/j.csi.2016.03.002.

[124] E. Sarmiento, J. C. S. P. Leite, E. Almentero, and G. S. Alzamora, "Test Scenario Generation from Natural Language Requirements Descriptions based on Petri-Nets," *Electron. Notes Theor. Comput. Sci.*, vol. 329, pp. 123–148, 2016, doi: 10.1016/j.entcs.2016.12.008.

[125] M. Steinberger, I. Reinhartz-berger, and A. Tomer, "Cross lifecycle variability analysis : Utilizing requirements and testing artifacts," *J. Syst. Softw.*, vol. 143, pp. 208–230, 2018, doi: 10.1016/j.jss.2018.04.062.

[126] K. Holder *et al.*, "Model-Based Requirements Management in Gear Systems Design Based On Graph-Based Design Languages," *Appl. Sci.*, vol. 7, no. 11, pp. 1–24, 2017, doi: 10.3390/app7111112.

[127] F. Mokammel, E. Coatanéa, J. Coatanéa, V. Nenchev, E. Blanco, and M. Pietola, "Automatic requirements extraction , analysis , and graph representation using an approach derived from computational," *Syst. Eng.*, vol. 21, no. 6, pp. 1–21, 2018, doi: 10.1002/sys.21461.

[128] M. Arias, A. Buccella, and A. Cechich, "A Framework for Managing Requirements of Software Product Lines," *Electron. Notes Theor. Comput. Sci.*, vol. 339, pp. 5–20, 2018, doi: 10.1016/j.entcs.2018.06.002.

[129] A. Albers and A. Kurrle, "Using semantic metadata for continuous development of requirements and goals in the Smart Mobility domain - An empirical study," *Procedia CIRP*, vol. 70, pp. 168–173, 2018, doi: 10.1016/j.procir.2018.03.070.

[130] I. Hajri, A. Goknil, L. C. Briand, and T. Stephany, "Change impact analysis for evolving configuration decisions in product line use case models," *J. Syst. Softw.*, vol. 139, pp. 211–237, 2018, doi: 10.1016/j.jss.2018.02.021.

[131] T. Wolfenstetter, M. R. Basirati, M. Böhm, and H. Krcmar, "Introducing TRAILS : A tool supporting traceability , integration and visualisation of engineering knowledge for product service systems development," *J. Syst. Softw.*, vol. 144, pp. 342–355, 2018, doi: 10.1016/j.jss.2018.06.079.

[132] C. Trubiani, A. Ghabi, and A. Egyed, "Exploiting traceability uncertainty between software architectural models and extra-functional results," *J. Syst. Softw.*, vol. 125, pp. 15–34, 2017, doi: 10.1016/j.jss.2016.11.032.

[133] H. Femmer, D. M. Fernandez, S. Wagner, and S. Eder, "Rapid quality assurance with Requirements Smells," *J. Syst. Softw.*, vol. 123, pp. 190–213, 2017, doi: 10.1016/j.jss.2016.02.047.

[134] A. Knauss, D. Damian, X. Franch, A. Rook, H. A. Müller, and A. Thomo, "ACon : A learning-based approach to deal with uncertainty in contextual requirements at runtime," *Inf. Softw. Technol.*, vol. 70, pp. 85–99, 2016, doi: 10.1016/j.infsof.2015.10.001.

[135] O. AlDhafer, I. Ahmad, and S. Mahmood, "An end-to-end deep learning system for requirements classification using recurrent neural networks," *Inf. Softw. Technol.*, vol. 147, no. January, p. 106877, 2022, doi: 10.1016/j.infsof.2022.106877.

[136] P. V. Nistala, A. Rajbhoj, V. Kulkarni, S. Soni, K. V. Nori, and R. Reddy, *Towards digitalization of requirements: generating context-sensitive user stories from diverse specifications*, vol. 29, no. 26. Springer US, 2022. doi: 10.1007/s10515-022-00324-2.

[137] A. Zaki-Ismail, M. Osama, M. Abdelrazek, J. Grundy, and A. Ibrahim, *RCM-extractor: an automated NLP-based approach for extracting a semi formal representation model from natural language requirements*, vol. 29, no. 10. Springer US, 2022. doi: 10.1007/s10515-021-00312-y.

[138] N. Handa, A. Sharma, and A. Gupta, "Framework for prediction and classification of non functional requirements: a novel vision," *Cluster Comput.*, vol. 25, no. 2, pp. 1155–1173, 2022, doi: 10.1007/s10586-021-03484-0.

[139] J. Frattini, J. Fischbach, D. Mendez, M. Unterkalmsteiner, A. Vogelsang, and K. Wnuk, "Causality in requirements artifacts: prevalence, detection, and impact," *Requir. Eng.*, 2022, doi: 10.1007/s00766-022-00371-x.

[140] H. Elhassan, M. Abaker, A. Abdelmaboud, and M. B. Rehman, "Requirements Engineering: Conflict Detection Automation Using Machine Learning," *Intell. Autom. Soft Comput.*, vol. 33, no. 1, pp. 259–273, 2022, doi: 10.32604/iasc.2022.023750.

[141] W. S. Jang and R. Y. C. Kim, "Automatic generation mechanism of cause-effect graph with informal requirement specification based on the korean language," *Appl. Sci.*, vol. 11, 2021, doi: 10.3390/app112411775.

[142] K. Kaur and P. Kaur, "SABDM: A self-attention based bidirectional-RNN deep model for requirements classification," *J. Softw. Evol. Process*, doi: 10.1109/RE.2007.45.

[143] A. Z. O. Salmon, P. M. G. del Foyo, and J. R. Silva, "A Formal Approach to Requirements Engineering of Automated Systems: Facing the Challenge for New Automated Systems," *J. Control. Autom. Electr. Syst.*, vol. 32, no. 4, pp. 815–829, 2021, doi: 10.1007/s40313-021-00731-y.

[144] Y. Zhang, J. Kang, and W. Dai, "Non-Functional Requirements Elicitation Based on Domain Knowledge Graph for Automatic Code Generation of Industrial Cyber-Physical Systems," in *IECON Proceedings (Industrial Electronics Conference)*, 2021, vol. 2021-Octob. doi: 10.1109/IECON48115.2021.9589564.

[145] A. P. Yanuarifiani, F.-F. Chua, and G.-Y. Chan, "Performance measurement of a rule-based ontology framework (ROF) for auto-generation of requirements specification," *Int. J. Inf. Technol. Syst. Approach*, vol. 15, no. 1, 2022, doi: 10.5220/0005329904030410.

[146] Z. Qi, T. Wang, and T. Yue, "Towards Efficient Use Case Modeling with Automated Domain Classification and Term Recommendation," in *Proceedings of the IEEE International Conference on Requirements Engineering*, 2021, vol. 2021-Septe, pp. 30–38. doi: 10.1109/REW53955.2021.00011.

[147] R. R. R. Merugu and S. R. Chinnam, "Automated cloud service based quality requirement classification for software requirement specification," *Evol. Intell.*, vol. 14, no. 2, pp. 389–394, 2021, doi: 10.1007/s12065-019-00241-6.

[148] A. Sleimi, N. Sannier, M. Sabetzadeh, L. Briand, M. Ceci, and J. Dann, "An automated framework for the extraction of semantic legal metadata from legal texts," *Empir. Softw. Eng.*, vol. 26, no. 3, 2021, doi: 10.1007/s10664-020-09933-5.

[149] M. Singh and G. Walia, "Automating Key Phrase Extraction from Fault Logs to Support Post-Inspection Repair of Software Requirements," 2021.

[150] N. K. Singh, M. Lawford, T. S. E. Maibaum, and A. Wassyng,

DOI: https://doi.org/10.29207/resti.v7i3.4924

"A formal approach to rigorous development of critical systems," *J. Softw. Evol. Process*, vol. 33, no. 4, 2021, doi: 10.1002/smr.2334.

[151] A. M. Maatuk and E. A. Abdelnabi, "Generating UML use case and activity diagrams using NLP techniques and heuristics rules," in *ACM International Conference Proceeding Series*, 2021, pp. 271–277. doi: 10.1145/3460620.3460768.

[152] Y. Elrakaiby, A. Borgida, A. Ferrari, and J. Mylopoulos, "CaRE: a refinement calculus for requirements engineering based on argumentation theory," *Softw. Syst. Model.*, no. 1, 2021, doi: 10.1007/s10270-021-00943-5.

[153] A. Henriksson and J. Zdravkovic, "Holistic data-driven requirements elicitation in the big data era," *Softw. Syst. Model.*, 2021, doi: 10.1007/s10270-021-00926-6.

[154] I. M. Subedi, M. Singh, V. Ramasamy, and G. S. Walia, "Application of back-translation - A transfer learning approach to identify ambiguous software requirements," in *ACMSE 2021: The Annual ACM Southeast Conference*, 2021, pp. 130–137. doi: 10.1145/3409334.3452068.

[155] J. Hochstetter, C. Díaz, M. Diéguez, and J. Díaz, "Proposal for a Classifier for Public Tenders for Software Based on Standard IEEE830," in *Communications in Computer and Information Science*, vol. 1444 CCIS, no. August, 2021, pp. 73–88. doi: 10.1007/978-3-030-84825-5_6.

[156] M. P. S. Bhatia, A. Kumar, and R. Beniwal, "An optimized classification of apps reviews for improving requirement engineering," *Recent Adv. Comput. Sci. Commun.*, vol. 14, no. 5, pp. 1390–1399, 2021, doi: 10.2174/2213275912666190716114919.

[157] M. S. Murtazina and T. V. Avdeenko, "An Approach to Extracting Ontology Concepts from Requirements," *Commun. Comput. Inf. Sci.*, vol. 1427, pp. 191–203, 2021, doi: 10.1007/978-3-030-81200-3_14.

[158] F. Bargui and H. Ben-Abdallah, "An ontology-based approach for automatic goal requirements engineering in data warehouse design," *Int. J. Inf. Decis. Sci.*, vol. 13, no. 2, pp. 143–165, 2021, doi: 10.1504/IJIDS.2021.116494.

[159] J. Fischbach *et al.*, "Automatic Detection of Causality in Requirement Artifacts: The CiRA Approach," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12685 LNCS, no. January, 2021, pp. 19–36. doi: 10.1007/978-3-030-73128-1_2.

[160] A. Schlutter and A. Vogelsang, "Trace Link Recovery Using Semantic Relation Graphs and Spreading Activation," in *Lecture Notes in Computer Science*, vol. 12685, 2021, pp. 37–53. doi: 10.1007/978-3-030-73128-1_3.

[161] J. Feng *et al.*, "FREPA: An automated and formal approach to requirement modeling and analysis in aircraft control domain," in *ESEC/FSE 2020 - Proceedings of the 28th ACM Joint Meeting European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2020, pp. 1376–1386. doi: 10.1145/3368089.3417047.

[162] M. S. Haris and T. A. Kurniawan, "Automated requirement sentences extraction from software requirement specification document," in *ACM International Conference Proceeding Series*, 2020, pp. 142–147. doi: 10.1145/3427423.3427450.

[163] Q. A. Do, T. Bhowmik, and G. L. Bradshaw, "Capturing creative requirements via requirements reuse: A machine learning-based approach," *J. Syst. Softw.*, vol. 170, p. 110730, 2020, doi: 10.1016/j.jss.2020.110730.

[164] S. Fritz, M. Jaenicke, J. Ovtcharova, and H. Wicaksono, "Context-sensitive Assistance in Requirements-based Knowledge Management," in *ACM International Conference Proceeding Series*, 2020, pp. 47–54. doi: 10.1145/3443279.3443306.

[165] A. Zaki-Ismail, M. Osama, M. Abdelrazek, J. Grundy, and A. Ibrahim, "RCM: Requirement capturing model for automated requirements formalisation," in *MODELSWARD 2021 - Proceedings of the 9th International Conference on Model-Driven Engineering and Software Development*, 2021, pp. 110–121. doi: 10.5220/0010270401100121.

[166] M. Osama, A. Zaki-Ismail, M. Abdelrazek, J. Grundy, and A. Ibrahim, "SRCM: A semi formal requirements representation model enabling system visualisation and quality checking," in *MODELSWARD 2021 - Proceedings of the 9th International Conference on Model-Driven Engineering and Software Development*, 2021, pp. 278–285. doi: 10.5220/0010271202780285.

[167] A. Zaki-Ismail, M. Osama, M. Abdelrazek, J. Grundy, and A. Ibrahim, "RCM-Extractor: Automated extraction of a semi formal representation model from natural language requirements," in *MODELSWARD 2021 - Proceedings of the 9th International Conference on Model-Driven Engineering and Software Development*, 2021, pp. 270–277. doi: 10.5220/0010270602700277.

[168] Y. Chen, S. Cao, G. Ma, and T. Li, "SAIP: A Prototype Tool for Semi-Automatic User Interface Prototyping," in *CEUR Workshop Proceedings*, 2021, vol. 2857.

[169] J. Fischbach, J. Frattini, and A. Vogelsang, "CiRA: A Tool for the Automatic Detection of Causal Relationships in Requirements Artifacts," in *CEUR Workshop Proceedings*, 2021, vol. 2857.

[170] R. K. Gnanasekaran, S. Chakraborty, J. Dehlinger, and L. Deng, "Using recurrent neural networks for classification of natural language-based non-functional requirements," in *CEUR Workshop Proceedings*, 2021, vol. 2857.

[171] M. Vidal, T. Massoni, and F. Ramalho, "A domain-specific language for verifying software requirement constraints," *Sci. Comput. Program.*, vol. 197, p. 102509, 2020, doi: 10.1016/j.scico.2020.102509.

[172] A. Gupta and C. Gupta, "CDBR : A semi-automated collaborative execute-before-after dependency-based requirement prioritization approach," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 2, pp. 421–432, 2022, doi: 10.1016/j.jksuci.2018.10.004.

[173] A. Zaki-ismail, M. Osama, M. Abdelrazek, J. Grundy, and A. Ibrahim, "Requirements Formality Levels Analysis and Transformation of Formal Notations into Semi-formal and Informal Notations," in *33rd International Conference on Software Engineering & Knowledge Engineering (SEKE 2021)*, 2021, pp. 1–6.

DOI: https://doi.org/10.29207/resti.v7i3.4924