

Towards Aspect Based Components Integration Framework for Cyber-Physical System

Sadia Ali¹, Yaser Hafeez¹, Muhammad Bilal², Saqib Saeed³ and Kyung Sup Kwak^{4,*}

¹University Institute of Information Technology, Pir Mehr Ali Shah Arid Agriculture University, Rawalpindi, 46000, Pakistan

²Department of Computer Engineering, Hankuk University of Foreign Studies, Yongin-si, 17035, Korea

³Department of Computer Information Systems, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, P. O. Box 1982, Dammam, Saudi Arabia

⁴Department of Information and Communication Engineering, Inha University, Incheon, 22212, Korea

*Corresponding Author: Kyung Sup Kwak. Email: kskwak@inha.ac.kr

Received: 21 March 2021; Accepted: 01 May 2021

Abstract: Cyber-Physical Systems (CPS) comprise interactive computation, networking, and physical processes. The integrative environment of CPS enables the smart systems to be aware of the surrounding physical world. Smart systems, such as smart health care systems, smart homes, smart transportation, and smart cities, are made up of complex and dynamic CPS. The components integration development approach should be based on the divide and conquer theory. This way multiple interactive components can reduce the development complexity in CPS. As reusability enhances efficiency and consistency in CPS, encapsulation of component functionalities and a well-designed user interface is vital for the better end-user's Quality of Experience (QoE). Thus, incorrect interaction of interfaces in the cyber-physical system causes system failures. Usually, interface failures occur due to false, and ambiguous requirements analysis and specification. Therefore, to resolve this issue semantic analysis is required for different stakeholders' viewpoint analysis during requirement specification and components analysis. This work proposes a framework to improve the CPS component integration process, starting from requirement specification to prioritization of components for configurable. For semantic analysis and assessing the reusability of specifications, the framework uses text mining and case-based reasoning techniques. The framework has been tested experimentally, and the results show a significant reduction in ambiguity, redundancy, and irrelevancy, as well as increasing accuracy of interface interactions, component selection, and higher user satisfaction.

Keywords: Cyber-physical systems; component-based development; case-based reasoning; prioritization; requirement management; specification; text mining



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1 Introduction

Software development is a complex activity that is human and knowledge-intensive [1,2]. The global market has enormous competition, quick and speedy changes in technology and tools. Thus, companies need to develop skills and resources to deliver cost effective, dependable software applications with higher user satisfaction in a short delivery time. The increasing complexity of the modern software applications and the need for a short time to market increased the size of the development team and other stakeholders. With the emergence of the global software development paradigm, large-scale collaboration became a strategic tool to survive in the market. The competitive organizations tend to involve partners and stakeholders who are globally distributed [3]. Global and locally connected networks facilitate collaboration among these global partners through software applications, hardware, and supporting processes [3–5]. Advancements in technological infrastructures introduce new challenges for software development processes and methodologies. The recent popularity of smart systems and the internet of things domains have highlighted the importance of Cyber-Physical Systems (CPS), which exhibit a complex interplay among hardware components, associated software, and related processes to fulfill requirements of end-users across different domains [6–9]. CPSs require real-time integration of different components to perform their assigned tasks in real-time environment [3–11]. We can categorize CPS components based on either organization or behavior of the components, as shown in Fig. 1.

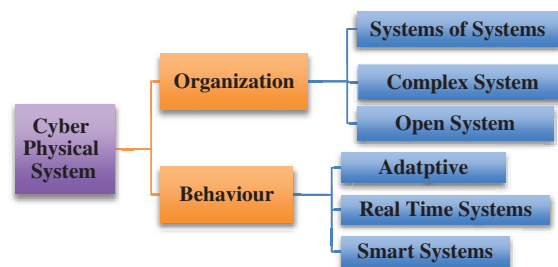


Figure 1: CPS structure

In the software engineering literature, the component-based software development (CBSD) approach has got the attention of industry and researchers in the last decade, which focuses on the composition of components as elementary blocks for product development [3,12–17]. Additionally, CBSD provides systematic reuse of the components which significantly improves the effectiveness of development teams. However, this can introduce interaction problems amongst configurable components because of term mismatches during the requirements management process, i.e., specification and prioritization activities [5,18–20]. Since CPSs are also a combination of different components and management of the CPSs components is prone to such problems in the development process. The typical phases of CPS product development are requirement management, design, and implementation as described in Fig. 2.

The complexity, redundancy, and conflicts in the requirements engineering phases of CPS components can result in the error propagation in all subsequent phases and increasing development time and cost. Requirements of any system are rooted in the needs of stakeholders' viewpoints, business needs, and operating environments of end-users. The requirements management phase is the important phase in the software development process, as requirements of systems are defined, documented, and maintained in this phase using natural language [4,21,22].

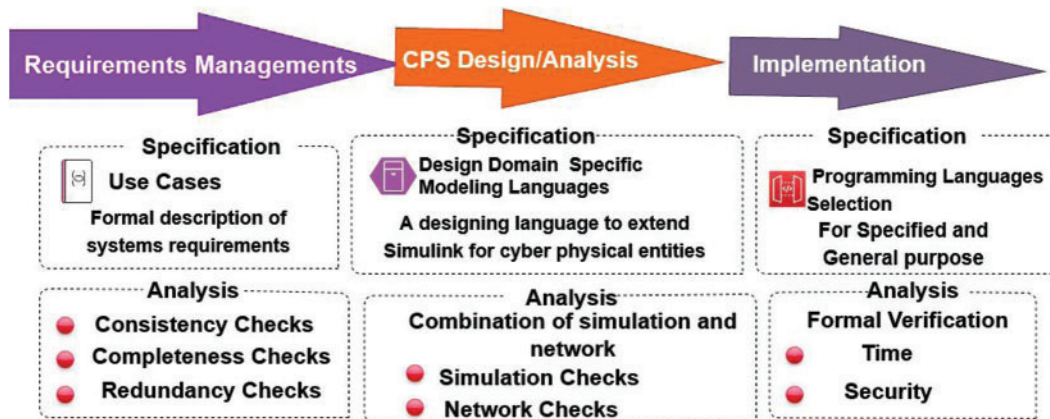


Figure 2: Phases of CPS development

The natural language raises the issues of redundancy, conflicts, and ambiguity, which may result in system failure at an early stage. These failures may trigger the server downfall at the time of configuration in CPS systems. Therefore, the requirement management process becomes more complicated during component specification and prioritization activities. In CBSD, proper and correct integration of components play a significant role, whereas, neglecting components, improper integration, and wrong interaction behavior of components, particularly in complex and configurable systems results in system failure [2,14,18,20–28].

CPS specification (CPSS) needs component configuration for the implementation during functionality changes. Consequently, after configuration, interactions of reused and new components become greatly difficult because of term mismatches as components are developed by various third parties aiming for different environments and expectations. CPS prioritization (CPSP) is the process of managing relative dependencies amongst components of CPS behaviors to cope with different functionality configurations within limited resources in complex projects. CPSP plays an important role in requirement management activities, particularly for critical tasks like requirements analysis and release [18,29]. To build and deliver good software that meets CPSP customer requirements, an effective software process is required to complete the job with preferential stakeholders’ requirements [30].

Several approaches can assist with CPSS and CPSP specifications based on stakeholder needs (cost, time, nature of the project, etc.). Most of these techniques are complex and may increase conflicts and redundancy by adopting different processes for CPSS and CPSP. The CPS development focuses on the integration of accurate and complete components to support the adoption of existing component’s interfaces. Consequently, existing techniques fail during the component integration phase due to improper CPSS and CPSP activities. In CPS, multi-users involvement having diverse perspectives and importance of components according to their needs results in misinterpretation and missing semantic information of multiple stakeholders’ viewpoints, requiring more efforts during CPSS and CPSP. Whereas selection of desirable components according to stakeholders’ needs require more human interaction and effort, resulting in system crash and resource shortage.

Therefore, there is a need for desirable system CPSS and CPSP processes in configurable CPS-based software where several interactions involve amongst components for higher satisfaction and reliability. For semantic, conflict, and redundancy analysis during the specification stage text

mining (TM) technique could be useful. The technique of extracting interesting and non-trivial patterns or information from unstructured text documents is known as text data mining or knowledge discovery from textual databases [29,31,32]. It's akin to data mining or information discovery from (structured) databases [29,31,32]. For knowledge management and reuse of previous knowledge, the researchers adapted case-based reasoning (CBR) as an artificial intelligence technique. CBR retrieves previous solutions for current problem-solving based on expert knowledge intelligently in different scenarios.

Our main contributions in this paper are as follows:

- Firstly, we present semantic-based CPS specification and prioritization framework using text mining and case-based reasoning, based on diverse users' viewpoints, managing reusability, and limited user involvement.
- Secondly, text mining is used for resolving ambiguous and conflicting requirements issues by extracting diverse stakeholder viewpoints semantically during configurable CPS specifications. We used two criteria current user priority and previous user ranking to prioritize CPS components after extraction of requirements semantically. CBR technique was used to extract previous similar used components ranking with less user's involvement to reduce stakeholders' conflicts. Therefore, the proposed framework resolves the drawback of configurable CPSS and CPSP processes using text mining and CBR.
- Thirdly, the framework is evaluated using experiment and analysis of the results highlight that this framework reduced ambiguity, and redundancy with higher satisfaction level to deal multi-viewpoints of stakeholders semantically and identifies reused requirements during CPS software development.
- Finally, the study offers a guide, a baseline, and empirical evidence for future research in the domain of continuous configuration management in CPS.

The rest of the paper is structured as follows: The second segment addresses similar work to illustrate current issues. The third section focuses on resolving the problems that have been found. In Section 4, we show the findings and analyze the suggested structure in conjunction with the results. Section 5 outlines the conclusion and future work of the report.

2 Related Work

Recently, CBSD is considered a more generalized approach for the CPS software development. To ensure the CPS quality, semantic-based specification and prioritization of configurable components are necessary. For high quality of CPS, someone must develop efficiently and effectively in the CBSD paradigm. For reliability of component requirements, most existing techniques focus on the post-integration phase of components. Still, few studies discuss specification and prioritization of components in the context of CPS development. The authors in [4] have proposed methods to enlist business and strategic requirements for a reconfigurable system.

The development process of the CPS is complex, and handling these complexities during the requirement engineering phase is a critical task. The [8] presented a requirements model for CPS, which provides guidelines about requirement refinement, collection, and clustering. They performed a case study about the application of the proposed model. However, there is still a need to focus on the semantic-based modeling using this requirements model [8]. The development process of CPSs requires close integration and vigilant coordination of many components. The [9] have focused on elicitation analysis and designing CPSs. There is still a need to rank and prioritize scenarios that are produced while performing the trade-off analysis procedure Previously available

requirements of security frameworks did not fulfill the needs of CPS security requirements. In [11], the authors proposed a framework for security requirements using an evolution approach and they evaluated this framework by applying it to a smart car parking system.

To realize high-quality, CPSs considering technological and service features are also important during the development process. Since such systems are complex and redundant, requirements for dynamic configuration to CPSS in RE for the product and service components are a significant issue. The author of [33] study provided a review of the CPSS definition and its implementation in an industrial survey to elucidate CPS engineering problems, focusing on the RE process. There is still a need to address the identified requirements with a CPSS RE framework. For the optimal development of CPS, creating a shared perception of the targeted CPS for the related stakeholders is necessary. The author of [22] used natural language processing to translate shared informal requirements to formal specification models. Still, there is a need to improve the semantic-based RE process to benefit CPS practical implementations. In [34], the authors proposed search-based software engineering for component selection and ranking is applied to produce results by using expert judgment. They automatically evaluated a set of components for a large telecommunications organization using a multi-objective greedy algorithm. They proposed a future recommendation to verify components by feature prioritization interactions. The [35] study prioritized components using object constraints language (OCL) to realize the system within time and other resource constraints. This approach enabled to reduce effort for the identification of faults-based components.

In the literature, there are few techniques for prioritization that satisfied specific quality criteria such as efficiency, scalability, and ease of use. The [31] proposed situation–transition structure method which required end-user involvement for requirement prioritization. In [36] study authors presented a technique for commercial off-the-shelf (COTS) prioritization for multi users' viewpoints. In [37] author presented systematic mapping and literature review to classify existing approaches to address selection and prioritization requirement problems. Similarly, [38] proposed a framework by using a fuzzy-based prioritization engine. In this approach, user prioritization value is used as an input with some fuzzy rules to benefits requirements analysis. Hence, [39] used a machine learning technique to deal with existing and new requirements priority orders. Based on users' feedback prioritization of requirements to reduced cost and time with less human effort [40]. In [41] proposed a combination of clustering and evolutionary-based algorithms to handle large data successfully using ranks. Thus, in literature scalability, accuracy, time consumption, etc. problems in the requirement prioritization process.

To identify term mismatch and semantic analysis, text mining methods are employed where Latent Semantic Index/Analysis (LSI/LSA) and Latent Dirichlet Allocation (LDA) concepts were implemented. For the independent review and audit of CPSS and CPSP requirements, the text mining approach was used to reduce quality assurance effort [4]. The approach used for similarity and dissimilarity of requirements investigated trace link assurance to reduce complexity.

Based on existing literature, we concluded that integrating CPS components into the CBSD process is a difficult and error-prone task. This is due to the lack of semantics and term mismatch problems resulting from the diverse views of multi-stakeholders throughout the definition of component specifications. This has an impact on all phases of the CPS development process, especially component prioritization activities. As a result, after a change, it increases uncertainty, human interaction, inconsistency, and ambiguity in configurable CPSs. Therefore, this paper propose a framework for improving configurable component specification and prioritization activities

using text mining and CBR for semantic and term mismatch, component ranking, and ranking predictions for similar cases problem of diverse stakeholders.

3 Framework for Requirement Management of Configurable CPS (RMCPs)

The configurable CPS requirement management process fails due to conflict, redundancy, and irrelevancy in requirements specification and prioritization which negatively impact other phases of CPS development. Therefore, this section proposes the RMCPs framework for CPS components requirements specification and prioritization using text mining and CBR techniques. The RMCPs framework provides comprehensive steps for configurable CPSS and CPSP for developing CPS based on semantic analysis, reusability of requirements and priority identification, and conflict removal for completeness.

The RMCPs framework considers diverse stakeholders' perspectives, less human interaction, reusability of requirements, and ranking of similar components of CPS for current CPS components and predicts missing ranking of selected components to resolve issues after configuration in complex configurable CPSs. Therefore, the RMCPs framework consists of three main steps i.e., requirement elicitation and analysis (REA), reusability manager, and prioritization as shown in Fig. 3.

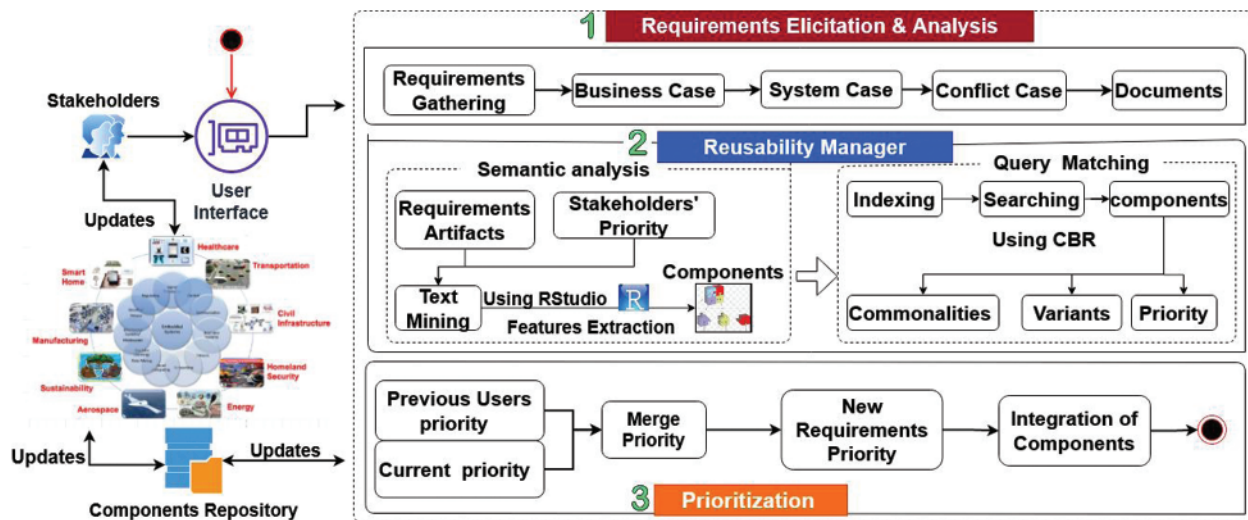


Figure 3: Requirement management of configurable CPS (RMCPs) framework

3.1 Requirement Elicitation and Analysis (REA)

In the REA phase, requirements are gathered using a web-based application from collaborating stakeholders and then collected we analyze requirements for the business case, system case, and conflict case for configurable CPS systems.

3.1.1 Business Case Requirements (BCR)

We categorize BCR requirements of CPS based on objectives, scope, benefits, performance, risks, roles, cost, resources, and rationales of the system. It helps to generate missing and incomplete requirements which are not collected during collaboration.

3.1.2 System Case Requirements (SCR)

In the SCR list, requirements relevant to response time, the volume of data, security, performance, usability, etc. are identified. These requirements of CPS may be conflicting and thus need to be managed carefully.

3.1.3 Conflict Case Requirements (CCR)

In CCR we focus the analysis on commonality and conflicts in requirements, thus leading to requirements merging and removal.

3.2 Reusability Manager (RM)

In the RM phase, requirements are structured, and someone identifies semantically reusable requirements for specification and prioritization of requirements. RM comprises two processes i.e., semantic analysis and query matching.

3.2.1 Semantic Analysis (SM)

In the SM process, we extract requirements from artifacts along with the priority of stakeholders of CPS using the RStudio tool (The tool automatically extracts terms semantically within and among all documents) for text mining (TM). TM is used to automatically analyze semantic information from the text in the form of terms based on the concept and their relationship. The K-nearest TM method is used for term extraction based on their frequencies. The following steps are used for TM [32,42,43]:

- Information extraction from CPS component specification terms.
- Eliminate stop words, prepositions, all repeated words, punctuation marks, etc.
- Remove plural into singular; removing “ing” from words, and words of similar context to find terms of CPS features.
- Extract CPS functionalities semantically to avoid inconsistency and incompleteness.

3.2.2 Query Matching (QM)

In the QM phase, all terms are indexed according to their frequency and search. Each CPS component using CBR for reusability of features and their priority for relevant and similar requirements to improve the prioritization process and reduce stakeholder involvement. CBR is attractive as it offers continuity and improves transparency with gained experience. CBR works on the reuse perception of a previous similar solution for requirement ranking to rank new CPS features and store ranking for future use in the central database [44–46]. The steps of CBR are:

- Retrieve components with similar functionalities: in this step, we match previous similar components with similar functionalities, and their ranking is saved in a repository with current functionalities using expert knowledge.
- Components adoption: In this phase, similar components are selected which match current components based on their previous ranking information.
- Reuse ranking: In this step, we reuse the previous ranking of stakeholders for similar components during components interaction in the integration process.

This is used to identify the ranking of missing current CPS components ranking to reduce human interaction and redundancy. During the elicitation process, some of the stakeholders are not directly involved and they use requirements after the completion of the development process.

3.3 Prioritization

In this phase, both current and previous priorities are merged to identify missing functionalities of some CPS components. This results in a new priority of semantically analyzed features of CPS components. It reduces incompleteness, inconsistency, conflicts, and ambiguity in feature priorities, due to less involvement of stakeholders. After this step, a list of priority of stakeholder components interaction prioritization is established, which is later sorted with a higher ranking of components. The higher priority components implementation for integration of desirable components facilitates stakeholders.

In the next section, based on these factors, we elaborate results of an empirical study with quantitative analysis, and this study verified that RMCPS enhanced requirements management activities by using CBR and text mining techniques.

4 Results and Discussion

In this section, we describe the results of the experiment performed to validate the activeness of RMCPS. In our experiment, we selected two projects i.e., Car Security Alarm and Patient record system of real-world software technologies company as case studies. The company used different methods for specification and prioritization to achieve higher user satisfaction and productivity requiring extensive human interaction. The consent of the participants for the evaluation of the proposed framework is acquired after the approval from the ethics committee of the selected organization using email and agreed to follow the organization's privacy policy about sharing the information about case studies and participants. Therefore, the evaluation design process, according to participants' knowledge and experience relevant to selected case studies. The participants of the said organization agreed to implement RMCPS to investigate user satisfaction and quality of product with proper CPSS and CPSP activities. We selected 12 participants and divided them equally into two groups, i.e., Experiment Treatment Group (ETG) and Control Treatment Group (CTG). The ETG group developed both projects using RMCPS, and CTG group adopted a traditional method for the development of both projects.

The participants included requirement analyst (RA), project manager (PM), Stakeholder (Sr), team leaders (TL), developers (Ds), and quality analyst (QA). After completing the project, we analyzed the progress based on some parameters which were identified from the existing literature for improving CPS component-based CPSS and CPSP i.e., easy to adopt (EA), component identification and retrieval (CIR), term mismatch resolves (TMR), semantic analysis (SA), increase productivity (IP), formal specification (FS), reduced human interaction (RHI), components prioritization process (CPP), prioritize desirable components (PDC), enhance components integration (ECI), proactive to changes (PTC), remove requirements conflict (RRC), remove requirements redundancy (RRR), increase process accuracy (IPC), increased completeness of requirements (ICR) and increased user satisfaction (IUS). Additionally, in the study we addressed the following research questions (RQs):

RQ 1: What is the effect of semantic-based requirement specification and prioritization on the outcome of the components integration process?

RQ 2: Does the implementation of RMCPS is produced better results than other relevant methodologies?

RQ 3: Can effectiveness of RMCPS improve the accuracy of the component's integration process.

To answer RQs, we experimented to extract parametric-based satisfaction levels. Therefore, the experiment starts with the first step of gathering the requirements of CPS-based data sets from the participants of experiments and map them with mentioned requirements. Then these requirements were divided as BCR, SCR, and CCR and documented. Then we used documents with their complete constraints and stakeholder viewpoints for semantic analysis using the RStudio tool. For example, in the case of Car security alarm, we extracted some of the terms after the text mining process which is listed in [Tab. 1](#).

Table 1: Extracted terms

S. No.	Requirements ID	Components functionalities
1	R-1	Door lock
2	R-2	Door unlock
3	R-3	Blink light
4	R-4	Activate alarm
5	R-5	Light blink

After extracting components functionalities, we extracted their current and previous ranking, as listed in [Tab. 2](#). Then in [Tab. 3](#) describes a comparison of priority results among RMCPS and traditional methods. As a result, we got a ranking of missing functionalities by reducing human interaction to avoid inconsistencies and ambiguities. Both methods have different results. [Tab. 4](#) presents factors analysis of members who contributed to the experiment. As shown by the results, most of the contributors were satisfied with the use of RMCPS as compared to those without using it. The results of both groups i.e., ETG and CTG contributors were reviewed based on factors depicted in [Figs. 4–6](#). The satisfaction level in [Figs. 4–6](#) shows contributors and factorial analysis on the y-axis and x-axis, respectively. At the same time, maximum participants in both projects have reported more than 50 percent satisfaction level. The ranking in [Tabs. 2](#) and [3](#) new ranking column explains that 5 is the highest-ranking level and less than 5 is the lowest level or less priority value of requirement.

Table 2: Extracted ranking

S. No.	Requirements ID	Current ranking	Previous ranking	New ranking	Sorting
1	R-1	5	4	5	R-1
2	R-2	–	4	4	R-5
3	R-3	0	3	3	R-4
4	R-4	1	3.5	3.5	R-2
5	R-5	4	4.5	4.5	R-3

Table 3: Ranking comparison

S. No.	RMCPS sorting	New ranking	Traditional method sorting	New ranking
1	R-1	5	R-2	5
2	R-5	4	R-3	4
3	R-4	3	R-1	3
4	R-2	3.5	R-5	2
5	R-3	4.5	R-4	1

Table 4: Factors analysis

Parameters	HS		S		N		DS		HD	
	ETG	CTG	ETG	CTG	ETG	CTG	ETG	CTG	ETG	CTG
EA	44	1	53	40	2	43	2	44	0	56
TMR	25	1	66	26	7	27	2	25	0	67
SA	36	1	56	38	6	37	2	36	0	58
FS	43	2	54	42	2	42	1	43	0	55
RHI	44	2	53	45	2	41	1	44	0	52
CIR	54	2	43	39	2	51	1	54	0	43
CPP	45	2	53	44	1	40	1	45	0	51
PDC	43	2	54	45	2	44	1	43	0	52
ECI	44	2	53	43	2	41	1	41	0	54
PTC	33	3	59	34	5	32	3	30	0	57
IP	34	3	58	37	5	29	3	33	0	59
RRC	54	2	43	39	2	51	1	54	0	43
RRR	45	2	53	44	1	40	1	45	0	51
ICP	43	2	54	45	2	44	1	43	0	52
ICR	44	2	53	45	2	41	1	44	0	52
IUS	30	2	60	33	8	32	2	31	0	61

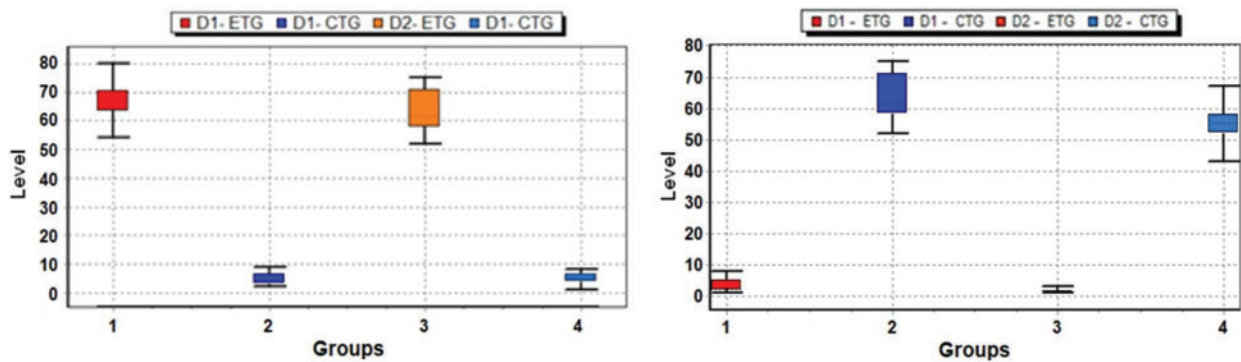


Figure 4: Highly satisfied and dissatisfied

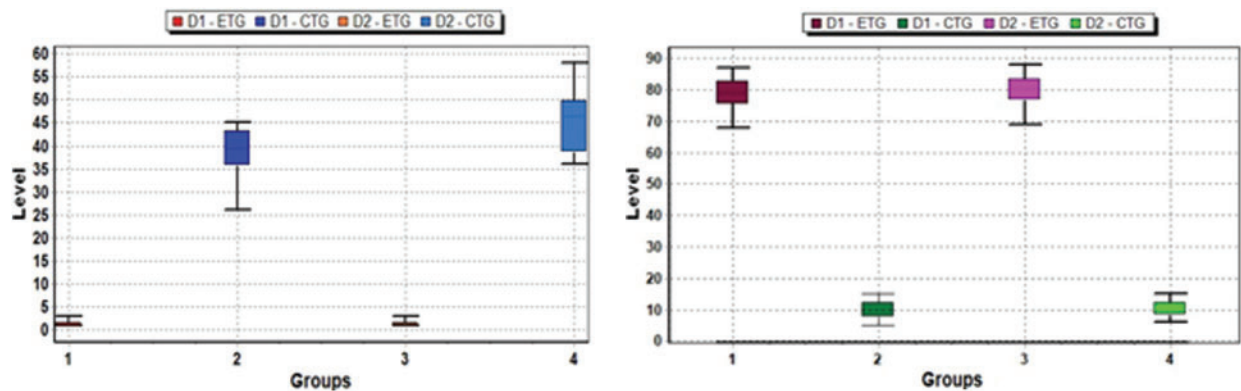


Figure 5: Neutral and satisfied

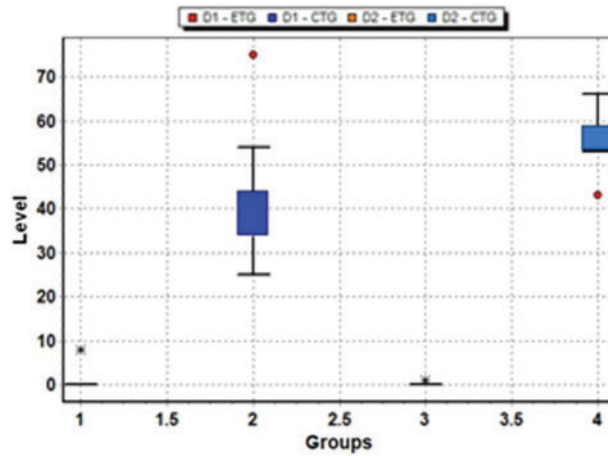


Figure 6: Highly dissatisfied

The rating scales used for evaluation are Highly satisfied (HS), Satisfied (Ss), Neutral, (Ns), Dissatisfied (Ds), and Highly dissatisfied (HD). The members of ETG who applied RMCPs were highly satisfied than CTG members who did not use RMCPs. The overall results show that RMCPs satisfaction for customer needs and quality yield good outcomes than without RMCPs.

To get answers to RQ2, we compared the results by statistical analysis with other techniques, i.e., Analytical Hierarchy Process (AHP). AHP prioritizes requirements in a pair-wise manner based on importance, penalty, cost, time, and risk, whereas the clustering method, divides a given set of data into several clusters to determine the relative closeness between those objects. Thus, for comparison of techniques, CTG adopted AHP and clustering method for requirement management process of integrated components to validate their functionalities. Tab. 5 depicts the overall satisfaction level of all the contributors after adopting RMCPs, AHP, and clustering techniques.

Table 5: Satisfaction level

Contributors	RMCPs	AHP	Clustering
RA	87	43	43
PM	84	51	53
Sr	65	52	54
TL	69	54	53
Ds	70	57	59
QA	72	59	58

For validating the questionnaire, we conducted a reliability test using the SPSS software 23 tool (To automatically calculate/solve reliability value of data for statistical analysis) to highlight the significant differences among the existing and our proposed RMCPs frameworks. For statistical analysis, we generated some null hypothesis (H) to check the reliability of collected data and the significance of RMCPs. These hypotheses are as follow:

H01: The TMR has no change using RMCPs, AHP, and Clustering methods.

- H02: The CIR has no change using RMCPS, AHP, and Clustering methods.
 H03: The RHI has no change using RMCPS, AHP, and Clustering methods.
 H04: The CPP has no change using RMCPS, AHP, and Clustering methods.
 H05: The PDC has no change using RMCPS, AHP, and Clustering methods.
 H06: The ECI has no change using RMCPS, AHP, and Clustering methods.
 H07: The PTC has no change using RMCPS, AHP, and Clustering methods.
 H08: The RRR has no change using RMCPS, AHP, and Clustering methods.
 H09: The RRC has no change using RMCPS, AHP, and Clustering methods.

The statistical results are shown in [Tab. 6](#) highlight that questions in the questionnaire were unbiased. To show the accuracy of RMCPS, we performed a statistical investigation using the SPSS tool. The t-statistical test was used for the analysis of reliability. The results explained that RMCPS appropriate as its t-value is less than the confidence interval (i.e., = 0.95).

Table 6: Reliability statistics

Cronbach's alpha	Cronbach's alpha based on standardized items	No. of items
0.911	1.00	12

To understand the implications and importance of various approaches like RMCPS, AHP, CV, we performed paired sample test on each of them and results. The results show diverse means of all groups, i.e., 0.72400 of RMCPS; while AHP and Clustering methods (i.e., 0.36236 and 0.47512 respectively) and interpreted that RMCPS values disperse less from their mean value and are more reliable than other methods, i.e., AHP and Clustering.

All tests have different t and means values which prove that experiment performance is reliable and unbiased without any ambiguity. As the results described, the mean value of the RMCPS approach is more than without RMCPS in both datasets, i.e., (0.728 and 0.744) and (0.356 and 0.370) respectively. Therefore, our proposed framework improves the process of specification and prioritization of configurable CBS.

Therefore, for RQ3 we used F-Measures and accuracy metrics [46,47] for accurate semantic analysis and correct selection of components for reusability. F-measure is a combination of both P and R, which shows the overall efficiency of the optimal test case selection process. It can be computed using [Eq. \(1\)](#).

$$F \text{ Measures} = (2 * P * R) / (P + R) \quad (1)$$

Precision (P) is the ratio of correctly specified and prioritized selected components to the total number of components available. Recall (R) is the ratio of components correctly specified and prioritized to the total number of components available for reusable. Accuracy (A) is defined as the ratio of correctly classified components to the total number of components. Accuracy is defined in [Eq. \(2\)](#).

$$A = (TP + TN) / (TP + TN + FP + FN) \quad (2)$$

TN shows similar components extracted for reusability. FP shows the number of components extracted for reusability but not selected. FN indicates the number of components extracted for

reusability but selected; while TP shows the number of components extracted for reusability and selected. The results of RQ are depicted in Fig. 7. In Fig. 7, values of metrics in percent are shown on the y-axis whereas the name of the metric is shown on the x-axis.

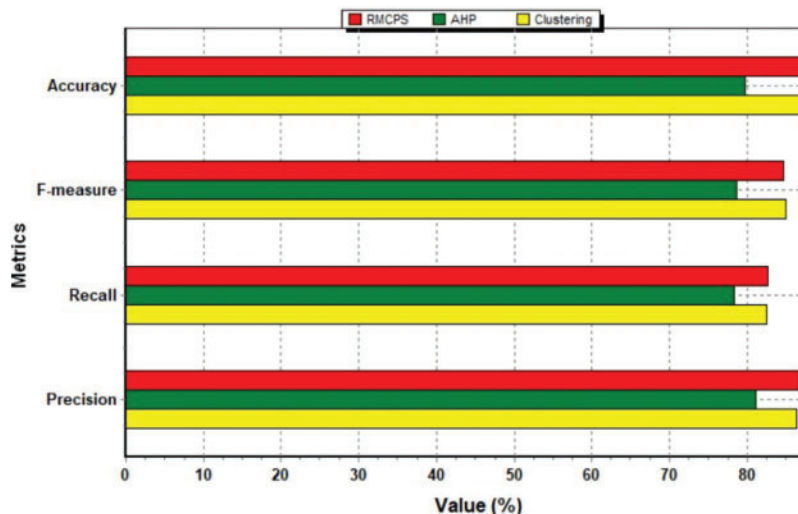


Figure 7: Values of metrics

The drawbacks of these case studies are that they are written in different programming languages, have different degrees of scalability, and have fewer expert participants available for text mining techniques. Consequently, in an empirical evaluation, many threats emerge that could cast doubt on the findings' theoretical rationality. Therefore, it is essential to repeat the study to accept or refute decisions. Internal validity (IV), external validity (EV), construct validity (CV), and reliability validity (RV) are the four main threats [47,48].

IV is concerned with considerations relating to the organization of requirements. To counteract this challenge, mitigation measures must be implemented to avoid the use of disparate measures for specification and prioritization. RMCPS strengthens the CPSS and CPSP processes, according to the findings of our research. In comparison to the used example for evaluation, EV refers to the generality of the results in actual projects. By replicating the PF measures in many cases, the validity of the findings is increased. The relationship between the different concepts and reflections is considered by CV. This necessitates the implementation of different criteria to determine the validity of various practices, such as semantic analysis and prioritization in PF to estimate output against other techniques. The relationship between behavior and effect is referred to as RV. This can be mitigated by performing a detailed review of the different decisions used in PF authentication. Data was collected by all the writers, and countermeasures were taken in the review of the findings. To avoid TV threats, we used an experiment to assess the learning effect, which may have affected the findings, as well as a qualitative study to minimize biases.

5 Conclusion and Future Work

Software development is a complex activity and with the advancements, in technological infrastructures, innovative practices need to be designed. With the recent popularity of CPSs, there is a need to optimize software engineering as well. To improve the requirement specification and

prioritization processes of highly configurable components, we proposed an RMCPS framework. This framework reduces the complexity of large set components reuse in interface interaction, and we employed semantic-based analysis for multi-user viewpoint satisfaction for CPS. We used text mining for specification and CBR to prioritize components to enhance the functionalities of products components with limited resources and high agility to reduce development time. To evaluate the effectiveness of our proposed framework, we experimented, and the results of the experiment showed a large mean difference (>0.5) and higher satisfaction (<0.5) in RMCPS as compared to traditional approaches. Thus, the proposed framework enhances the CPSS and CPSP activities and provides a roadmap for researchers and industry practitioners in the domain of CPS specification and prioritization activities. As future work, we intend to extend our proposed framework in a globally distributed environment to resolve specification and prioritization issues. Furthermore, we intend to mitigate the issue of quality analysis after continuous modification in CBSD in the cloud computing environment as well.

Funding Statement: This work was supported by National Research Foundation of Korea-Grant funded by the Korean Government (Ministry of Science and ICT)-NRF-2020R1A2B5B02002478).

Conflicts of Interest: The authors declare that they have no conflicts of interest with this study.

References

- [1] S. Yadav and B. Kishan, "Analysis and assessment of existing software quality models to predict the reliability of component-based software," *International Journal of Emerging Trends in Engineering Research*, vol. 8, no. 6, pp. 371–380, 2020.
- [2] D. Badampudi, K. Wnuk, C. Wohlin, U. Franke, D. Smite *et al.*, "A decision-making process-line for selection of software asset origins and components," *Journal of Systems and Software*, vol. 135, pp. 88–104, 2018.
- [3] C. Ayala, "System requirements-Oss components: Matching and mismatch resolution practices—An empirical study," *Empirical Software Engineering*, vol. 23, no. 6, pp. 3073–3128, 2018.
- [4] F. Boschi, C. Zanetti, G. Tavola and M. Taisch, "Functional requirements for reconfigurable and flexible cyber-physical system," in *IECON 2016-42nd Annual Conf. of the IEEE Industrial Electronics Society*, Florence, Italy, pp. 5717–5722, 2016.
- [5] V. T. Heikkilä, M. Paasivaara, C. Lasssenius, D. Damian and C. Engblom, "Managing the requirements flow from strategy to release in large-scale agile development: A case study at ericsson," *Empirical Software Engineering*, vol. 22, no. 6, pp. 2892–2936, 2017.
- [6] M. U. Tariq, M. Babar, M. A. Jan, A. S. Khattak, M. D. Alshehri *et al.*, "Security requirement management for cloud-assisted and internet of things—enabled smart city," *Computers, Materials & Continua*, vol. 67, no. 1, pp. 625–639, 2021.
- [7] M. Wajeeh, Y. Hafeez, S. Hussain, H. Anwaar, S. Yang *et al.*, "Recommender system for configuration management process of entrepreneurial software designing firms," *Computers, Materials & Continua*, vol. 67, no. 2, pp. 2373–2391, 2021.
- [8] Y. Lu and M. R. Asghar, "Semantic communications between distributed cyber-physical systems towards collaborative automation for smart manufacturing," *Journal of Manufacturing Systems*, vol. 55, pp. 348–359, 2020.
- [9] J. Ayerdi, A. Garcíandia, A. Arrieta, W. Afzal, W. Enoiu *et al.*, "Towards a taxonomy for eliciting design-operation continuum requirements of cyber-physical systems," in *2020 IEEE 28th Int. Requirements Engineering Conf.*, IEEE, Zurich, Switzerland, pp. 280–290, 2020.
- [10] U. Markiegi, "Test optimisation for highly-configurable cyber-physical systems," in *Proc. of the 21st Int. Systems and Software Product Line Conf.-Volume B*, Sevilla, Spain, pp. 139–144, 2017.

- [11] S. U. Rehman and V. Gruhn, "An effective security requirement engineering framework for cyber-physical systems," *Technologies*, vol. 6, no. 3, pp. 1–65, 2018.
- [12] F. A. Bukhsh, A. Bukhsh and M. Daneva, "A systematic literature review on requirement prioritization techniques and their empirical evaluation," *Computer Standards & Interfaces*, vol. 69, pp. 103389, 2020.
- [13] I. Gambo, R. Ikono, P. Achimugu and A. Soriyan, "An integrated framework for prioritizing software specifications in requirements engineering," *Requirements Engineering*, vol. 12, no. 1, pp. 33–46, 2018.
- [14] M. Hussain, M. Ahmed, H. A. Khattak, M. Imran, A. Khan *et al.*, "Towards ontology-based multi-lingual URL filtering: A big data problem," *Journal of Supercomputing*, vol. 74, no. 10, pp. 5003–5021, 2018.
- [15] B. I. Ya'u and M. N. Yusuf, "Building software component architecture directly from user requirements," *International Journal of Engineering and Computer Science*, vol. 7, no. 2, pp. 23557–23566, 2018.
- [16] B. Graics, V. Molnár, A. Vörös, I. Majzik and D. Varró, "Mixed-semantics composition of statecharts for the component-based design of reactive systems," *Software and Systems Modeling*, vol. 19, no. 6, pp. 1483–1517, 2020.
- [17] E. Stachtari, A. Mavridou, P. Katsaros, S. Bliudze and J. Sifakis, "Early validation of system requirements and design through correctness-by-construction," *Journal of Systems and Software*, vol. 145, pp. 52–78, 2018.
- [18] M. Arias, A. Buccella and A. Cechich, "A framework for managing requirements of software product lines," *Electronic Notes in Theoretical Computer Science*, vol. 339, pp. 5–20, 2018.
- [19] E. Bjarnason, H. Sharp and B. Regnell, "Improving requirements-test alignment by prescribing practices that mitigate communication gaps," *Empirical Software Engineering*, vol. 24, no. 4, pp. 2364–2409, 2019.
- [20] L. Kaur and A. Mishra, "Software component and the semantic Web: An in-depth content analysis and integration history," *Journal of Systems and Software*, vol. 125, pp. 152–169, 2017.
- [21] K. Lace and M. Kirikova, "Required changes in requirements engineering approaches for socio-cyber-physical systems," in *REFSQ Workshops*, Utrecht, The Netherlands, pp. 1–17, 2018.
- [22] S. Wiesner, C. Gorltd, M. Soeken, K. D. Thoben and R. Drechsler, "Requirements engineering for cyber-physical systems," in *IFIP Int. Conf. on Advances in Production Management Systems*, Springer, Berlin, Heidelberg, pp. 281–288, 2014.
- [23] I. Gonzalez-Herrera, J. Bourcier, E. Daubert, W. Rudametkin, O. Barais *et al.*, "Scapegoat: Spotting abnormal resource usage in component-based reconfigurable software systems," *Journal of Systems and Software*, vol. 122, pp. 398–415, 2016.
- [24] T. Olsson, K. Wnuk and T. Gorschek, "An empirical study on decision making for quality requirements," *Journal of Systems and Software*, vol. 149, pp. 217–233, 2019.
- [25] A. Sebaa and A. Tari, "Query optimization in cloud environments: Challenges, taxonomy, and techniques," *The Journal of Supercomputing*, vol. 75, no. 8, pp. 5420–5450, 2019.
- [26] A. Shatnawi, A. D. Seriai, H. Sahraoui and Z. Alshara, "Reverse engineering reusable software components from object-oriented APIs," *Journal of Systems and Software*, vol. 131, pp. 442–460, 2017.
- [27] T. Vale, I. Crnkovic, E. S. De Almeida, P. A. D. M. S. Neto, Y. C. Cavalcanti *et al.*, "Twenty-eight years of component-based software engineering," *Journal of Systems and Software*, vol. 111, pp. 128–148, 2016.
- [28] H. Xie, J. Yang, C. K. Chang and L. Liu, "A statistical analysis approach to predict user's changing requirements for software service evolution," *Journal of Systems and Software*, Vol. vol. 132, pp. 147–164, 2017.
- [29] S. Ali, M. Imran, Y. Hafeez, T. R. Abbasi, W. Haider *et al.*, "Improving component based software integration testing using data mining technique," in *2018 12th Int. Conf. on Mathematics, Actuarial Science, Computer Science and Statistics (MACS) IEEE*, Karachi, Pakistan, pp. 1–6, 2018.
- [30] A., Alzaqebah R. Masadeh and A. Hudaib, "Whale optimization algorithm for requirements prioritization," in *2018 9th Int. Conf. on Information and Communication Systems (ICICS) IEEE*, Irbid, Jordan, pp. 84–89, 2018.

- [31] N. L. Atukorala, C. K. Chang and K. Oyam, "Situation-oriented evaluation and prioritization of requirements," In: S. W. Lee, T. Nakatani (Eds.), *Requirements Engineering Toward Sustainable World*, vol. 671. Singapore: Springer, pp. 18–33, 2016.
- [32] Z. Tong and H. Zhang, "A text mining research based on LDA topic modelling," in *Int. Conf. on Computer Science, Engineering and Information Technology*, Canada, pp. 201–210, 2016.
- [33] S. Wiesner, E. Marilungo and K. D. Thoben, "Cyber-physical product-service systems—challenges for requirements engineering," *International Journal of Automation Technology*, vol. 11, no. 1, pp. 17–28, 2017.
- [34] P. Baker, M. Harman, K. Steinhofel and A. Skaliotis, "Search based approaches to component selection and prioritization for the next release problem," in *2006 22nd IEEE Int. Conf. on Software Maintenance*, Philadelphia, PA, USA, pp. 176–185, 2006.
- [35] A. Jalila and D. J. Mala, "Software components prioritization using OCL formal specification for effective testing," in *2013 Int. Conf. on Recent Trends in Information Technology (ICRTIT)*, Chennai, India, pp. 714–720, 2013.
- [36] R. Santos, A. Albuquerque and P. R. Pinheiro, "Towards the applied hybrid model in requirements prioritization," *Proc. Computer Science*, vol. 91, pp. 909–918, 2016.
- [37] A. M. Pitangueira, R. S. P. Maciel and M. Barros, "Software requirements selection and prioritization using SBSE approaches: A systematic review and mapping of the literature," *Journal of Systems and Software*, vol. 103, pp. 267–280, 2015.
- [38] S. Dhingra, G. Savithri, M. Madan and R. Manjula, "Selection of prioritization technique for software requirement using fuzzy logic and decision tree," in *2016 Online Int. Conf. on Green Engineering and Technologies (IC-GET)*, Coimbatore, India, pp. 1–11, 2016.
- [39] R. Naseem, Z. Shaukat, M. Irfan, M. A. Shah, A. Ahmad *et al.*, "Empirical assessment of machine learning techniques for software requirements risk prediction," *Electronics*, vol. 10, no. 2, pp. 149–168, 2021.
- [40] F. Shao, R. Peng, H. Lai and B. Wang, "DRank: A semi-automated requirements prioritization method based on preferences and dependencies," *Journal of Systems and Software*, vol. 126, pp. 141–156, 2017.
- [41] P. Achimugu and A. Selamat, "A hybridized approach for prioritizing software requirements based on K-means and evolutionary algorithms," in *Computational Intelligence Applications in Modeling and Control*, Springer, Cham, pp. 73–93, 2015.
- [42] P. U. Hatipoğlu, A. Atvar, Y. O. Artan, O. Şereflişan and A. Demir, "Software requirement traceability analysis using text mining methods, in *2017 25th Signal Processing and Communications Applications Conf. (SIU)*, Antalya, Turkey, pp. 1–4, 2017.
- [43] D. Port, A. Nikora, J. H. Hayes and L. Huang, "Text mining support for software requirements: traceability assurance," in *2011 44th Hawaii Int. Conf. on System Sciences*, Kauai, HI, USA, pp. 1–11, 2011.
- [44] A. Smiti and Z. Elouedi, "Dynamic maintenance case base using knowledge discovery techniques for case-based reasoning systems," *Theoretical Computer Science*, vol. 817, pp. 24–32, 2020.
- [45] Y. Li, T. Yue, S. Ali and L. Zhang, "Enabling automated requirements reuse and configuration," *Software & Systems Modeling*, vol. 18, no. 3, pp. 2177–2211, 2019.
- [46] S. Albahli, A. Algsham, S. Aeraj, M. Alsaeed, M. Alrashed *et al.*, "Covid-19 public sentiment insights: A text mining approach to the gulf countries," *Computers, Materials & Continua*, vol. 67, no. 2, pp. 1613–1627, 2021.
- [47] Y. Hafeez, S. Ali, N. Jhanjhi, M. Humayun, A. Nayyar *et al.*, "Role of fuzzy approach towards fault detection for distributed components," *Computers, Materials & Continua*, vol. 67, no. 2, pp. 1979–1996, 2021.
- [48] B. Miranda and A. Bertolino, "An assessment of operational coverage as both an adequacy and a selection criterion for operational profile-based testing," *Software Quality Journal*, vol. 26, no. 4, pp. 1571–1594, 2018.