Journal of Computer & Robotics 15 (2), Summer and Autumn 2022, 1-14

Computer
& Robotics

RCMS: Requirements Conflict Management and Overlapping Control Strategy in CSOP+RP using Pearson correlation coefficient

Soheil Afraz[a], Hassan Rashidi[b]*, Naser Mikaeilvand[c]

[a] Faculty of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

[b] Department of Mathematics and Computer Science, Allameh Tabataba'i University, Tehran, Iran

[c] Department of Mathematics, Ardabil Branch, Islamic Azad University, Ardabil, Iran

Received 07 August 2022; Accepted 13 September 2022

Abstract

Requirement engineering is one of the critical phases in the software development process. Functional Requirements (FR) and Non-Functional Requirements(NFR) are two of the fundamental requirements in software projects that are observed in the classifications of most researchers in the software engineering field. Conflicting and overlapping among the requirements in both intra and extra communications levels are one of the main challenges in the elicitation and prioritization phases. This paper presents a decision strategy to respond to this challenge called requirements conflicts management strategy (RCMS). This strategy is defined to manage conflict and overlap of NFRs in the prioritization of the constraints satisfaction model for requirements prioritization, known as "CSOP + RP" model. In this strategy, the necessary constraints and RCMS is applied to the "CSOP+RP" model as a pre-processing phase by the requirement analyzer and the results are delivered to the system manager. In the definition of RCMS, they are several components: the conflicts catalog among NFRs, the mapping model of NFRs to the domain of software systems, and the Pearson correlation coefficients in NFRs. The correlation coefficients are calculated on the importance of the requirements, which mean conflict, overlap and neutral, respectively. To evaluate RCMS, it was implemented on Police Command-and-Control System(PCCS) as a case study with specific NFRs and FRs. The statistical analysis of the experimental results shows that the proposed strategy increases the accuracy of the input values of the prioritization model and better decision-making in managing conflicts and controlling overlaps. Furthermore, RCMS help to reduce the ambiguities between NFRs and FRs and also influences of NFRs in requirement ranking by the search-based prioritization approach.

Keywords: Conflicts Management, Functional Requirements, Non-Functional Requirements, Overlapping control, Prioritization model, Pearson correlation coefficients, Strategy.

1.Introduction

Clear definitions of Non-Functional requirements(NFRs) and Functional Requirements (FRs) are essential factors for the success of software projects because they focus on the main problem of software quality. NFRs attend to conflict with each other, and this conflict is one of the critical features of NFRs. Multiple models of conflicts for NFRs are proposed, and the interactive nature of NFRs is a positive or negative internal relation among NFRs. The positive relation represents a pair of NFRs supporting each other's while the negative relation represents a contradiction between NFRs.

Conflict (negative effect) and overlapping (positive effect) between requirements, and especially NFRs

* Corresponding Author. Email: hrashidi@qiau.ac.ir

are among the most critical challenges in elicitation and prioritization of requirements. This challenge domain of systems, the collaboration of stakeholders in requirement organization and different views of the system, nature conflict among several requirements (explicit conflict), different relations and dependencies, and necessity of trade-off considering of nature of systems. Ignoring this vital item could negatively affect requirement prioritization and lead to wrong analysis and invalid results for developing different versions or the next software release in the software product line.

The conflict handling and management process must consider identifying, analyzing, and resolving the requirements. Three approaches that can be classified as manual, automatic, and semi-automatic are proposed for each step. The aggregation of the proposed classifications for conflict management consists of the anthological approaches, methodological and technical approaches on the one hand, and formal approaches, modeling and using stakeholder preferences, especially in aspect-oriented requirements engineering, on the other hand [23].

In recent years, the artifacts in requirement conflict and overlapping research areas have focused on tools generation, general and specific frameworks (depending on the domain and case studies of the system), metaheuristic approaches (search-based), and the development and automation of previous approaches. Therefore, conflict management and influencing control in requirements prioritization increase certainty and clarity in the final ranking of requirements.

The presented solution in the model "CSOP + RP" [49] is a decision strategy RCMS based on proportional variation and applying specific constraints in the mathematical model of the problem along with the mapping and replacing it in the general schema. In this paper, Afraz et al. (2021) proposed a decision-oriented methodology with a novel model for requirements prioritization (RP) in large-scale software systems. The model was formulated based on the constraint satisfaction optimization problems (CSOP) approach, known as depends on many factors such as the variety of context and application CSOP+RP. The main objective of the model was to maximize the quality of the software in total, subject to the constraints on the budgets and importance level pre-determined by the administrator. To evaluate CSOP+RP, the authors applied it to the Police Command-and-Control System (PCCS), which is extensively used during the outbreak of the Coronavirus disease as an incident in terms of quality and speed of service. The results of various experiments showed that the proposed model, with its specific capabilities, could find a near-optimal solution. Moreover, the sensitivity analysis indicated that the model is very sensitive to its parameters. The CSOP+RP is very general, so it can be applied to different types of software projects.

Therefore, the RCMS decision strategy is presented based on creating relative changes and applying specific constraints in the mathematical model of the problem(CSOP+RP). The replacement of RCMS in the proposed methodology is the prioritization of requirements with a statistical approach based on the calculation of the correlation coefficient, which is detailed in the following sections.

The remaining sections are as follows. In Section2, the literature review, the existing techniques are classified into three categories manual, automatic and semi-automatic, and the general frameworks with different approaches are analyzed. The preliminaries of the research contain the concept of the Pearson correlation coefficient, and the basic definitions are stated in the third section. The proposed strategy (RCMS) with a statistical approach for conflict management and overlapping control of the requirements based on the defined components and their steps is explained in Section 4. The application of the RCMS strategy to the PCCS case study is described in the fifth section, and the implementation results are shown. In section 6, statistical analysis and discussion of the results are based on the derived charts. Finally, in Section 4, the results and the main challenges in this area are highlighted.

2. Literature Review

We did an extensive study on the approach/method used for conflict management, conflict management type, type of requirements, the scope of the approach/method used, description, and requirement representation. The results of this study are summarized in Tables 1,2, and 3. In these tables, the 'Reference', 'Approach/Method used for Conflict Management', 'Conflict Management Type', 'Type of Requirements', 'Scope of the Approach/Method used', 'Description and Requirement Representation', and 'Publication Year' are specified, respectively, in its columns from left to right. According to the information in the tables, we classified the existing techniques of previous research work into (a) automatic and semi-automatic (Table 1), (b) manual (Table 2), and (c) general frameworks (Table 3). In this section, the methods are reviewed briefly.

2.1. Automatic and Semi-Automatic Techniques

Egyedand Grunbacher used a traceability technique for conflict resolution and wrong collaborations in the automatic and semi-automatic classes [26]. This technique resolved the requirements automatically to identify the requirements with conflicting attributes. The traceability resolution automatically identified the traceability dependencies among requirements and detected the overlaps. Kim et al. [9] introduced a systematic and automatic process to detect and manage the requirements conflicts according to requirements partitioning in source and activity types in the natural language by the RECOMA tool. Kamalrudin et al. [8] described the traceability approach used to manage the consistency between textual and abstract interaction requirements and the essential applications. Moser et al. [13] proposed a semantic approach for requirements conflict detection.

Escalona et al. ([19], [20]) proposed a model-based approach to identify requirements conflicts in web applications. Nguyen et al. [27] proposed a knowledge-based requirements Engineering (KBRE) based on domain knowledge and semantics. Chentouf proposed an automatic formalization method for FR requirement conflict management called validation rules [21].

Hausmann et al. [12] worked on conflict identification of FRs based on UML and using the use-case diagram for FR identification. Gervasi and Zowghi [37] conducted conflict inference of requirements in natural language based on logic using theorem proving and model checking, as proposed in [44] and [45], and considering some stakeholders' points of view.

Fletcheck and Lin presented a new model for service selection based on fuzzy logic, which considers users' personal preferences and interactions with NFR attributes during service selection [34].

In 2016, Liu provided an automatic technique called CDNFRE for evolutionary NFR conflict detection based on ontology as a fundamental theory to support ONFRA conflict resolution [38]. This technique performed the conflict resolution and detecting process in four steps, apriori knowledge modeling, new NFRs modeling, side effect identification, and conflict detection, based on four essential components; metadata, ontology, conflict detection rules, and causal relations.

Mala and Uma presented an approach to identifying NFRs by describing a particular use case based on domain models such as UML class diagrams and goal-oriented questionnaires [39]. This approach used a domain model to extract the system behavior and possible restrictions for its actors. Therefore, NFR ranking and user precedence were used to resolve them according to interaction resolution based on the prioritization of precedence in NFR. The interface engine performed prioritization, and the precedence was calculated from specified user weights for each NFR.

In 2019, Zang and Wang proposed a conflicting NFRS trade-off framework (CNTF) framework [48]. The proposed framework operates based on a trade-off approach, and the stakeholders' importance amount of NFRs is stated using linguistic variables. Shah et al. [40], in 2019, proposed an ontological approach to identify formal conflicts among NFRS. This approach aims to formally identify conflicting NFRs from natural language NFRs based on ontological representation. It helps requirement analysts to detect and prioritize NFRs using meta-model, semantic catalog, and Semantic Web Rule Language (SWRL). Additionally, they invented the intra-conflicts concept in 2021 [47]. They proposed a semi-automatic approach using natural language processing, machine learning, and ontological-based semantic analysis to identify the conflicts among NFRs.

2.2. Manual Techniques

In the class of manual techniques, Heisel and Souquires presented a metaheuristic algorithm to detect the conflict of attributes and features in the requirements [3]. In this algorithm, the requirements for conflict were selected, and then the software engineers and stakeholders performed the final detection. Robinson [6] used root requirements analysis in which a composition of three procedures is applied: first, the requirements are written in a structural form, the hierarchy of root requirements is generated, and finally, the root requirements are analyzed based on their expected degree of conflicts to sort the requirements.

Poort et al. [15] proposed a Non-functional Decomposition (NFD) which considered the NFR to FR mapping for architecture design. They established a modal and iterative framework for decomposing a conflict-based system in system requirements. Sadana and Liu[16] proposed a framework to resolve the conflicts among NFRs using the uniform analysis of FR and NFR. Liu [29] applied an ontological approach for conflict analyses in requirement specifications on the activity diagram.

Heng and Ming introduced a non-mathematical technique called "multi-coordinated view" that represents different stakeholder views [17]. In 2010, Mairiza and Zowghi [5] proposed an ontological framework called SureCM to manage the conflicts between security and usability requirements. Butt et al. [2] applied the MEO strategy to resolve requirements conflicts using requirements filtering and analysis. This strategy filtered the extracted requirements in the elicitation phase into mandatory, essential, and optional requirements based on their implementation nature. Then it is analyzed and resolved based on the Conflict Resolving Strategy (CRS).

In 2013, Mairiza and Zawghi [11] introduced a practical approach to design an ontological framework that managed the relative conflicts among NFRs. In 2014, Mairiza and Zawghi [24] proposed a novel idea for NFR conflict resolution based on multi-criteria decision-making. It is called TOPSIS in which the quantity effects of NFRs on FRs were identified according to the stakeholders' weighting statements, and the FR Ranking was performed based on the defined steps and relations.

Abraham et al. [25] presented a structural method to detect the conflicts among candidate function requirements in three phases. In the first phase, all conflicts were removed after a diagram analysis of the problem. Then in the second phase, the candidate conflict requirements were reduced using some information about parallel or nonparallel completion of the requirements. Finally, in the third phase, the candidate conflicts were controlled and reduced if the preconditions were composed and completed.

Merilinna et al. [35] developed the SIG framework, which was introduced by Chung et al. [41]. This framework aimed to quantify the NFRs and bidirectional traceability of NFRs without automatic inference. Using a goal-oriented approach, Hu et al. [36] semantic modeling and automatic reasoning of NFRs conflicts in the context of Softgoal interdependencies.

2.3. General Frameworks

This class is neither manual nor automatic. Shehata et al. [22] introduced a three-level framework to detect conflicts. The first level is an informal approach to helping experts and the second level is a semiformal and semantic way without formal methods. The third level applies formal approaches to detect conflicts precisely.

In 2011, Miriza and Zowghi proposed a framework that makes a catalog to identify the NFR conflicts [28]. In his thesis, Boehm implements two prototyped supporting tools called QARCC and S-COST that extend the capabilities of the Win-Win model based on negotiation [43]. QARCC focuses on software architecture strategies to reach the aims of quality attributes, and S-COST addresses the interactions among software cost, functional and other quality features.

Carvalho presented a catalog of conflicts among NFRs, considering UniComp and IoT as novel NFRs during design time in executable models [42]. In management type, type of requirement, scope of the approach, description and requirement representation, and publication year is specified for each work.

3. Preliminaries

In this section, we review some of the research's preliminaries comprising the Pearson correlation coefficient and related basic definitions in the proposed strategy (RCMS).

3.1.Pearson Correlation Coefficient

The Pearson correlation coefficient is a measure of the linear dependence (degree of linear correlation) between two random variables (real-valued vectors). The definition of the Pearson correlation coefficient is as follows [50]:Definition: Assume two samples X and Y which can be denoted as vectors: $\vec{X}$ and $\vec{Y}$. Each sample contains N sample observations which can be denoted as the components of the vectors, namely, $\vec{X}$= [x_1, x_2, …, x_N] and $\vec{Y}$= [y_1, y_2, …, y_N]. Then the Pearson correlation coefficient of $\vec{X}$ and $\vec{Y}$ is calculated according to (Eq.1).

Table1, the conflict management approach, conflict

$$r_{\vec{X}\vec{Y}} = \frac{N\sum_{i=1}^{N} x_i y_i - \sum_{i=1}^{N} x_i \sum_{i=1}^{N} y_i}{\sqrt{N\sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i\right)^2}\sqrt{N\sum_{i=1}^{N} y_i^2 - \left(\sum_{i=1}^{N} y_i\right)^2}} \quad (1)$$

The main properties of Pearson correlation coefficient are that:

(i) The value range of is $r_{\vec{X}\vec{Y}}$ [−1, 1].
(ii) If $r_{\vec{X}\vec{Y}} > 0$, $\vec{X}$ and $\vec{Y}$ is positive correlation.
(iii) If $r_{\vec{X}\vec{Y}} = 0$, the linear correlation of $\vec{X}$ and $\vec{Y}$ is not obvious.
(iv) If $r_{\vec{X}\vec{Y}} < 0$, $\vec{X}$ and $\vec{Y}$ is negative correlation.
(vi) The greater $|r_{\vec{X}\vec{Y}}|$ is, the higher linear correlation rate of $\vec{X}$ and $\vec{Y}$ will be.

It should be noted that the Pearson correlation coefficient is for handling the linear relationship of two variables. A nonlinear correlation coefficient should be used if the relationship is nonlinear, such as Spearman's rank correlation coefficient [51]and Kendall's rank correlation coefficient [52].

Table 1
Automatic/ Semi-Automatic type of previous works in the field of requirement conflict management and overlapping control

Reference No. / publication Year	**Approach/Method used for Conflict Management**	**Type of Requirements**	**Scope of the Approach/Method**	**Description and Requirement Representation**
[26] / 2004	Traceability approach	Functional Non-functional	Identify	
[9] / 2007	Requirements partition in natural language Detecting conflicts in goal and scenario	Functional and Partially Considering NFR	Identify Analyze (source conflicts, activity conflict)	Formalization, Configurable rule support Not disclosed
[8] / 2010	Tractability approach	Functional	Identify	Structure model (EUC)
[13],[18] / 2011	Semantic-based approach	Functional	Identify Analyze (CRC, CRG,	Ontology
[20] / 2012 [19] / 2013	Graphical method using NDT meta model	Functional	Identify Analyze	Formalization(DSL) Structure model(NDT requirement Meta model)
[27] / 2013	Graphical method using requirement goal graph	Functional	Identify Analyze	Ontology Formalization(OWL) Structure model
[21] / 2014	Validation rules	Functional	Identify Analyze (7 types)	Formalization
[12] / 2002	Detecting conflicts in UML use case	Considering NFR Not disclosed		Configurable rule support Not disclosed
[37] / 2005	Detecting inconsistencies in if…then statement	Considering NFR Not disclosed		Configurable rule support Not disclosed
[34] / 2015	Using tradeoff strategies to select services with conflicting non-functional attributes	Partially Considering NFR , software attributes		Configurable rule support Not disclosed
[38] / 2016	Ontology-based Approach, CDNFR	Non-functional (software attribute, business value, restriction, and evaluation are	Identify Analyze Resolve	Lack of formalism and conflict detection rules, no details about the classification of conflicts, the rule editor module and rule base are provided
[39] / 2006	Goal-Oriented and UML-based approach: user preferences and NFR taxonomy to analyze the trade-offs among NFRs as the result of conflict and cooperation among NFRs	Non-functional	Identify Modeling	identify the non-functional requirements for a given use case description from the domain model such as Unified Modelling Language class diagram and goal based questionnaires
[40] / 2019	ology-based Approach	Non-functional	Identify Specification	Configurable rule supported, formally specify conflicting NFRs from available natural language NFRs by means of ontological representation
[47] / 2021	Detecting Intra-Conflicts	Non-functional	Identify Analyze	The analyst uses natural language processing, machine learning, and
[48] / 2019	Tradeoff analysis	Non-functional	Analyze (Tradeoff)	Trade-off decisions for conflicting NFRs based on Linear Programming

Note: Except for [40], [47], and [48] works (Semi-Automatic), other works are Automatic type.

Table 2
Manual type of previous works in the field of requirement conflict management and overlapping control

Reference No. / publication Year	**Approach/Method used for Conflict Management**	**Type of Requirements**	**Scope of the Approach/Method**	**Description and Requirement Representation**
[3] /2001	Heuristic algorithm	Functional	Identify	Formalization (schematic versions)
[6] / 2004	Root requirements analysis	Functional Non-functional	Identify Analyze (different	Formalization (schematic versions)
[15] / 2004	Non-functional decomposition Model (NFD)	Non-functional	Identify Analyze (grouping group conflicts) Resolve conflicts.	
[16] / 2007	Integrated analysis of FRs and NFRs	Non-functional	Identify Analyze(mutually exclusive, partial)	Formalization (two canonical form is developed)
[29] / 2009	Model-based UML activity diagram	Functional	Identify Analyze (7 types of conflicts)	Structure model (Activity Diagram)
[17] / 2010	Non-mathematical technique	Functional	Identify Analyze (3 types) Resolve	Formalization (semi-formal ontology driven domain-special requirement language)
[5] / 2010	Ontological framework	Non-functional (security and usability)	Identify Analyze (natural of conflict) Resolve	Ontology
[2] / 2011	MEO-strategy	Functional Non-functional	Identify Analyze (mandatory, essential, optional)	
[11] / 2013	Experimental approach using NFRs metrics and measures as parameters	Non-functional	Identify Analyze (strong, weak)	
[24] / 2014	A goal-based technique (TOPIS)	Non-functional	Resolve	
[30] / 2014	Graphical method using problem diagram	Functional	Identify	Structure model (problem diagram)
[35] / 2015	NFR+ Framework, Goal-Oriented Approach, traceability	Non-functional	Identify & specification	Extended SIG [41], Difficult UML integration, No automation, Measurable NFRs reasoning
[36] / 2015	Ontology-based approach, Semantic web representation of NFR Framework approach	Non-functional	Modeling	Manual SIG representation, no analysis based on the standard quality model, No details about the classification of conflicts

Table 3
previous works in the field of requirement conflict management and overlapping control in type of General-Framework

Reference No. / publication Year	**Approach/Method used for Conflict Management**	**Type of Requirements**	**Scope of the Approach/Method**	**Description and Requirement Representation**
[22] / 2004	Three-level interaction detection framework	Functional	Identify	
[28] / 2011	Investigation of research on NFRs and build a catalog of NFRs conflicts	Non-functional	Analyze (absolute, relative, no conflict)	
[43] / 2001	Using Win-Win quality requirements management tools: a case study	Functional and Non-functional	Identify Analyze Resolve	
[42]/ 2017	Dealing with Conflicts between Non-functional Requirements	Non-functional	Identify Analyze	Considering UbiComp and IoT Applications

3.2. Coefficient Basic definitions in RCMS

The important concepts in the proposed strategy (RCMS) are Conflict, Overlap and Neutral, which are explained below:

- **Conflict:** Conflicts between non-functional requirements (NFRs) mean that achieving one requirement can impact another. This can negatively affect the effective factors in ranking requirements (such as importance, cost, risk, etc.) in the development of software systems (negative effect). Therefore, the final results are incorrect in the requirements prioritization based on the mutual influence of these two NFRs.

- **Overlap:** Unlike the conflict concept, the positive and aligned effect of two NFRs on ranking factors in the prioritization of requirements is known as overlap (Positive effect). The overlapping of two NFRs will not impact the results of prioritizing requirements, but it should be controlled against conflicts.

- **Neutral (No effect):** If there is no conflict or overlap between two NFRs, it is called a neutral state. Due to the lack of effect of this state on the final results of prioritizing the requirements, it is considered with the overlapping state in the proposed strategy.

4. Requirement Conflicts Management Strategy (RCMS)

Considering the mathematical model of requirement prioritization in the CSOP+RP model [49], the RCMS strategy is designed and defined in this section. Additionally, it used the relative effect of dependencies based on opinions of Decision-Making Units(DMUs) that have a relatively structural analysis and identification. This strategy operates based on the following three components:

(I). The knowledge of domain mapping for various types of NFRs is divided into five common domains of software systems, as shown in Figure 1. These domains are (a) Real-time systems; (b) Safety Critical systems; (c) Web systems; (d) Information systems and (e) Process control systems. The knowledge helps to identify some explicit conflicts of NFRs in each domain's shared and specific states.

(II). The conflicts catalog among NFRs (Figure 2) is according to the nature of conflicts and overlaps. They are used to identify explicit conflicts in different types, such as absolute (label X), relative (label *), and no-conflict (label 0) [28].

(III). The positive and negative effects between quality attributes for the interaction in requirement engineering and software architecture as Figure 3[46].

Considering the above components and the decision parameters as well as the variables of the primary model of the proposed approach, the steps of the RCMS strategy in the proposed model "CSOP+RP" are as follows:

- **Step1-** Assignment the impact factor from the requirements to the others based on the DMU opinions.

- **Step2-** Averaging the DMU views according to the weight defined by the manager for every DMU in various systems.

- **Step3-** Calculate the Pearson correlation coefficient to manage conflicts and control the overlap of NFRs based on statistical approach.

5. Application of RCMS in CSOP+RP

As an application of the proposed strategy in software systems, RCMS was applied to the dataset of a designed case study.

5.1.Case Study (PCCS)

The designed case study, namely PCCS (Police Command-and-Control System) is a real-time system implemented in the CSOP+RP model. Briefly, 10 NFRs and 13 FRs have been considered in four subsystems: registration, Decision-Making, Dispatching, and Reporting. The detail of these NFRs and FRs in the PCCS system and their abbreviations and notations can be seen in Table 4 and Table 5.

Table 4
The subsystem of CCPS with its functional requirements

Subsystems	Notation	FR Requirement (Abbreviation.)
Registration	FR_1	Call taking (CT)
	FR_2	Incident Registration (IR)
Decision Making	FR_3	Create Response (CR)
	FR_4	Find Closest Unit (FU)
	FR_5	Alert Emergency Service (ES)
	FR_6	Get Position of Units (GP)
Dispatching	FR_7	Dispatch Units (DU)
	FR_8	Send Data (SD)
	FR_9	Response to Incident (RI)
	FR_{10}	Incident/Unit management(IM)
	FR_{11}	Request more units (RU)
Reporting	FR_{12}	Sending Report (SR)
	FR_{13}	Closing Incident (CI)

Table 5
Non-Functional requirements in the CCPS system

Notation	NFR Requirement (Abbreviation.)
NFR_1	Security (SY)
NFR_2	Performance (PF)
NFR_3	Availability (AV)
NFR_4	Accessibility (AC)
NFR_5	Usability (US)
NFR_6	Reliability (RL)
NFR_7	Maintainability (MT)
NFR_8	Interoperability (IO)
NFR_9	Scalability (SC)
NFR_{10}	Portability (PO)

5.2.Experimental Results

According to the designed dataset for PCCS case study, we use the importance value of requirements to the RCMS and evaluate it. Therefore, based on the acquired data from DMUs (DMU views) in the proposed methodology for CSOP+RP [49] in the form of average weight of the importance of the NFRs on the FRs (Table 6), the RCMS was implemented. The results of calculating the Pearson correlation coefficient (the third step of RCMS) on NFR pairs in PCCS. This table contains three positive, negative, and zero values, which are interpreted as values of overlap, conflict, and neutral (no effect) situations in the proposed strategy, respectively. Considering the three key components of RCMS, i.e., the application domain of the software system and extracting its NFRs (Figure 1), the catalog of conflicts of NFRs (Figure 2), and the effects of NFRs (Figure 3), the system administrator analyzed the values of Figure 4. The rows and columns of the table in Figure 4 are the candidate NFRs of the system. Also, Pearson's correlation coefficient values are placed in Figure 4. The heatmap's color spectrum corresponds to high overlap with white color, neutral status by red color, and highest conflict with black color between the NFRs. Therefore, the system administrator can make the best decision for conflict management and overlap control of NFRs based on heatmap results.

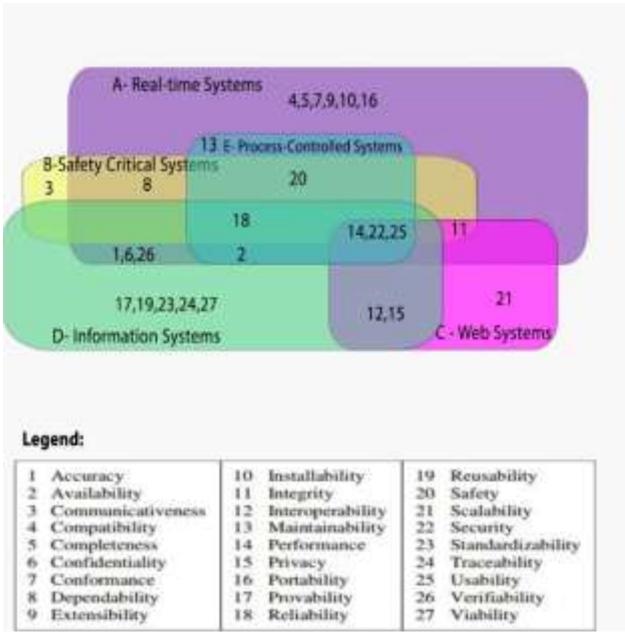

Fig. 1. The NFRs map to different types of software systems based on the application domain

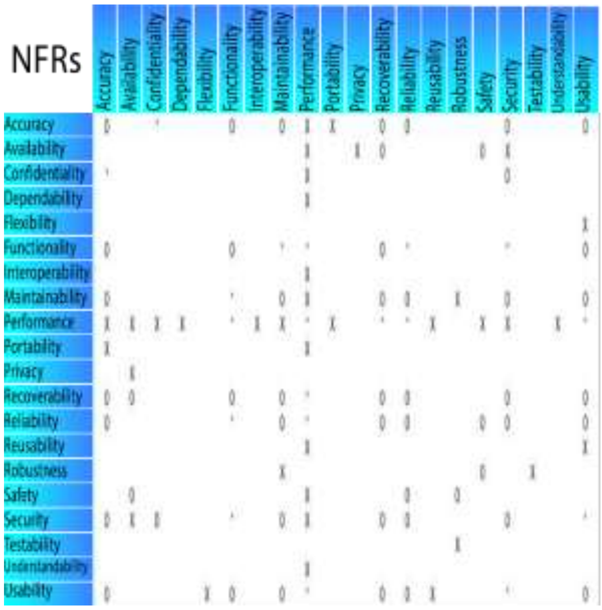

Fig. 2. The catalog of conflicts among NFRs

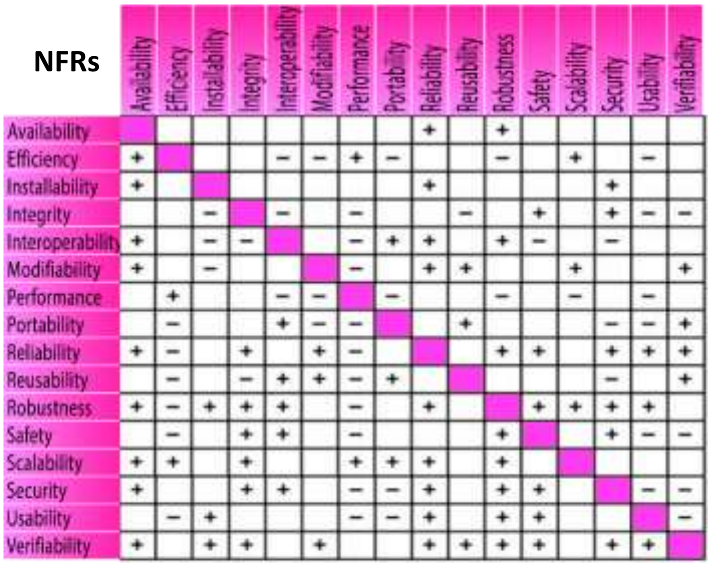

Fig. 3. The positive and negative effects between selected quality attributes

Table 6
The average weight of importance of the NFRi on the FRj based on DMU views

	FR1	FR2	FR3	FR4		FR5	FR6	FR7	FR8	FR9	FR10	FR11	FR12	FR13
NFR1	4	4	2	2		3	3	4	5	4	4	3	5	4
NFR2	4	5	4	5		3	5	5	4	4	5	4	5	4
NFR3	3	3	4	3		2	3	3	4	3	2	2	3	2
NFR4	5	2	2	2		2	3	3	3	4	3	4	4	2
NFR5	3	3	2	5		2	2	4	3	2	2	3	4	2
NFR6	3	4	3	3		2	2	3	3	3	1	3	3	2
NFR7	2	3	2	1		1	1	2	3	3	2	2	1	2
NFR8	1	2	1	4		4	1	1	1	2	1	4	2	1
NFR9	2	4	4	3		4	2	3	3	2	2	3	1	1
NFR10	2	1	2	2		3	1	1	2	1	1	3	2	1

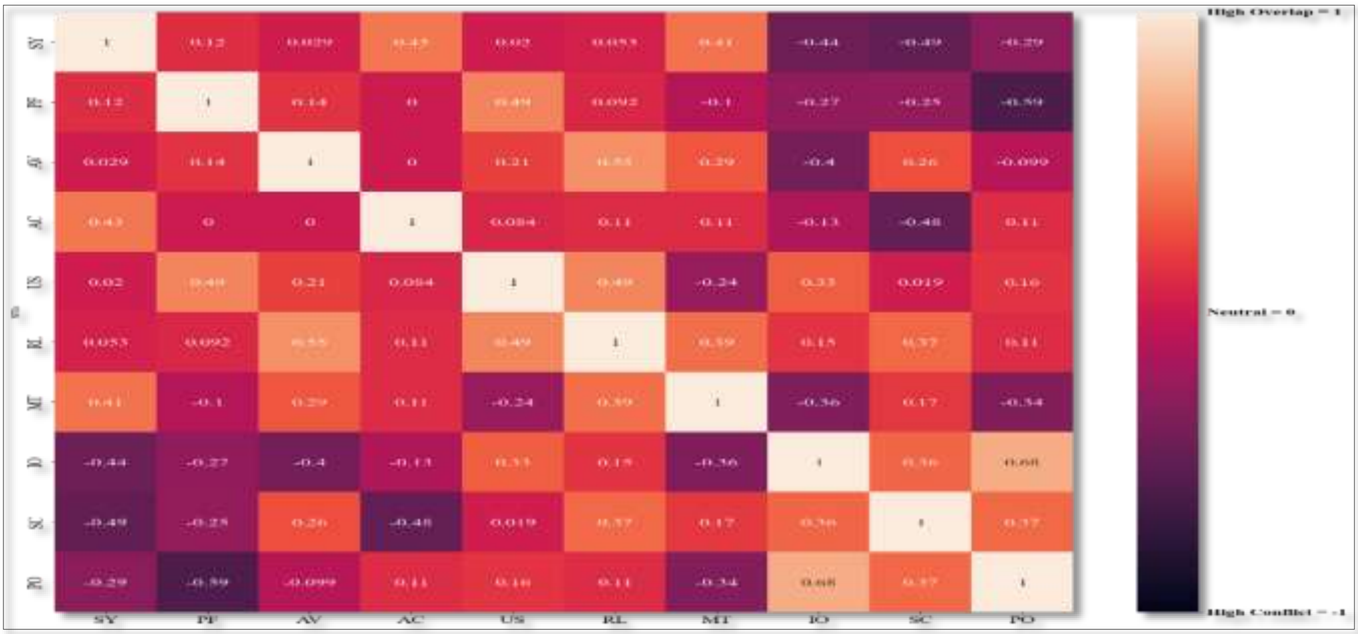

Fig.4. Heatmap of Pearson correlation coefficients on PCCS NFRs

6. Analysis and Discussion

In this section, from the point of view of the system administrator, the results of the RCMS implementation in the CSOP prioritization model are analyzed and discussed on the PCCS case study system.
Due to the nature of the designed case study and the time sensitivity in its operation, PCCS falls in the real-time system category (type A in Figure 1). Considering to the area specified for real-time systems, they are mapped to designated NFRs. According to the importance of some of these NFRs as well as the constraints defined in the system (such as cost), the system administrator in PCCS has selected 10 NFR of them and averaged the weight of the view of DMUs. Now, at this stage, based on the results of RCMS implementation on PCCS dataset, by considering the important NFRs and analyzing their mutual effect on the views obtained on the FRs, the system administrator should make the appropriate decision to manage conflicts and control the overlap of requirements.

Based on the statistical analysis and interpretation of sample data in figure 4 and figure 5 is derived. Besides the logical values in figure 4 and the symmetry of the values of the upper triangle and the lower triangle, we ignore the value of the diagonal elements of the correlation coefficient matrix, which are equal to 1 (since they are the correlation of an NFR with itself). Therefore, we can obtain the following observations in sample analysis.

- **Observation1 (Conflict Analysis):** According to Figure 5, the highest number of conflicts have occurred for Interoperability (IO) (see Fig. 5-a), based on the input data in Table 6, these conflicts (including SC and PO) are mainly focused on the FRs of the Decision-Making(DM) subsystem and should be considered by the admin. In addition, Security(SC) is a crucial quality attribute in the system (see Fig. 5-b) and conflicts with IO, SC and PO, directly affecting all subsystems. Therefore, the administrator must have special control over the importance values of FRs in the Registration and Reporting subsystem. Similarly, Performance (PF) also conflicts with MT, IO, SC, and PO (see Fig. 5-c), which must be managed due to the high importance values of the NFR on the system FRs.

- **Observation2 (Overlap Analysis):** As seen in Fig. (5-d), the Reliability(RL) completely overlaps with others. This shows that the specified importance values in all subsystems are correct from the point of view of DMUs and will not negatively impact the final prioritization of system FRs. The Usability(US) is another NFR that overlaps with other NFRs except for MT, which is the most overlapping with PF and RL (see Fig. 5-e).

- **Observation3 (Neutral Analysis):** According to Fig. (5-f), the Accessibility(AC) is in a neutral status with PF and AV symmetrically. According to the definition of section 1, the status is considered to be in the same direction as the overlap status. As a result, there is no need to change the values of DMU views and handling by the administrator for prioritization.

7. Conclusion and the challenges

As quality attributes, identifying and managing conflicts and controlling the overlaps among NFRs is one of the important challenges in requirement elicitation and prioritization phases. This impact is due to the variety of requirements and their cross effects on software systems' internal and external communication levels, especially the Ultra Large-Scale software systems (ULSS).

This paper proposes a strategy based on a composition approach for conflict management and overlapping control as RCMS on the NFRs. This strategy is driven by several essential components defined. Based on the defined operational steps, RCMS as a pre-processing phase on the CSOP+RP model (as a general model) can be implemented on different domains of software systems. Using the Pearson correlation coefficient on NFRs is known as contribution of this research. Pearson's correlation coefficient values correspond to the states of conflict, overlap and neutral and are analyzed by the system administrator.

The statistical analysis of experimental results indicates that the implementation of RCMS on PCCS (as a designed case study) reduces ambiguities the results of using this strategy. As a result, the relative accuracy of the input of the prioritization model has reached its highest level and the ambiguities in the final prioritization list are reduced.

According to the importance of the research area, several critical problems and major challenges have existed in the field of conflict management and overlapping control among requirements emphasizing scalability such as:

- The lack of solid meta-data and certainty in identification rules with requirement analysis in knowledge databases corresponds to the conflict management process.
- The lack of specified structure, syntax, and the ambiguity in requirement definitions and their in requirements prioritization. Therefore, the system administrator or other stakeholders can better manage conflicts and control overlaps by analyzing attributes, especially NFRs (Quality attributes) based on different interpretations and applications of Natural Language Processing (NLP) methods.
- The lack of a standard dictionary and catalog of NFRs that could be approved by software research associations and global software institutes, according to the variety of software systems domains and the development methodologies.
- Inconsistency in stakeholder precedencies based on the needs, and various views with different goals and expectations of the system.
- The challenges of software architecture design, according to the lack of specified definitions of the NFRs and their attributes.

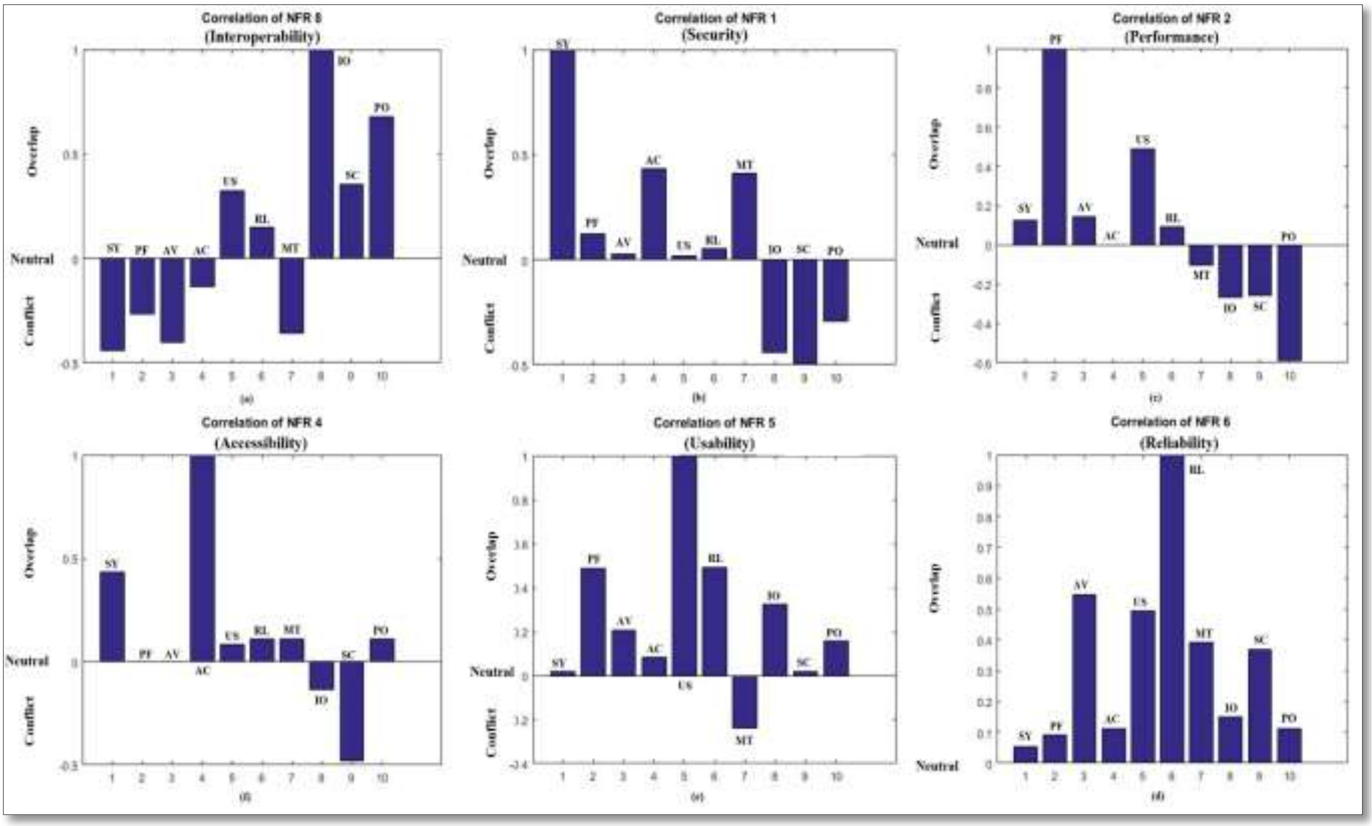

Fig. 5. Conflict, overlap and neutral status on NFRs of PCCS system separately

References

[1] D. Mairiza, D. Zowghi, and N. Nurmuliani, Managing conflicts among non-functional requirements, University of Technology, Sydney, 2009.

[2] W. H. Butt, S. Amjad, and F. Azam, Requirement Conflicts Resolution: Using Requirement Filtering and Analysis in Computational Science and Its Applications - ICCSA 2011.

[3] M. Heisel and J. Souquires, A Heuristic Algorithm to Detect Feature Interactions in Requirements in Language Constructs for Describing Features, S. G. Bs. (Hons) and M. R. B. MA, Eds. Springer London, 2001, pp.143-162.

[4] M. Ramzan, Intelligent Requirement Prioritization using Fuzzy Logic, Ph.D., National University of Computer and Emerging Sciences, Pakistan, 2010.

[5] D. Mairiza and D. Zowghi, An ontological framework to manage the relative conflicts between security and usability requirements in 2010 Third International Workshop on Managing Requirements Knowledge (MARK), 2010, pp.1-6.

[6] W. N. Robinson, Surfacing Requirements Interactions in Perspectives on Software Requirements, J. C. S. do P. Leite and J. H. Doorn, Eds. Springer US, 2004, pp. 69-90.

[7] B. Schar, Requirements Engineering Process HERMES 5 and SCRUM, University of Applied Sciences and Arts, Northwestern Switzerland, 2015

[8] M. Kamalrudin, J. Grundy, and J. Hosking, Managing Consistency between Textual Requirements, Abstract Interactions and Essential Use Cases in Computer Software and Applications Conference(COMPSAC), 2010 IEEE 34th Annual, 2010, pp.327-336.

[9] M. Kim, S. Park, V. Sugumaran, and H. Yang, managing requirements conflicts in software product lines: A goal and scenario based approach, Data Knowl. Eng., vol. 61, no. 3, pp. 417-432, Jun.2007.

[10] W. N. Robinson, S. D. Pawlowski, and V. Volkov, Requirements Interaction Management, ACM ComputSurv, vol. 35, no. 2, pp.132-190, Jun.2003.

[11] D. Mairiza, D. Zowghi, and V. Gervasi, Conflict characterization and Analysis of Non-Functional Requirements: An experimental approach, in 2013 IEEE 12th International Conference on Intelligent Software Methodologies, Tools and Techniques (SoMeT), 2013, pp. 83-91.

[12] Hausmann, Jan Hendrik, and Heckel, Reiko and Taentzer, Gabi, Detection of conflicting functional requirements in a use case-driven approach: a static analysis technique based on graph transformation in Proceedings of the 24th international conference on software engineering, ACM, 2002, pp.105-115.

[13] T. Moser, D. Winkler, M. Heindl, and S. Biffl, Requirements Management with Semantic Technology: An Empirical Study on Automated Requirements Categorization and Conflict Analysis in Advanced Information Systems Engineering, H. Mouratidis, and C. Rolland, Eds. Springer Berlin Heidelberg, 2011, pp. 3-17

[14] M. Shehata, A. Eberlein, and A. Fapojuwo, IRIS: a semi-formal approach for detecting requirements interactions in Engineering of Computer-Based Systems, 2004. Proceedings. 11th IEEE International Conference and Workshop on the, 2004, pp. 273-281.

[15] E. R. Poort and P. H. N. de With, Resolving requirement conflicts through non-functional decomposition in Fourth Working IEEE/IFIP Conference on Software Architecture, 2004. WICSA 2004. Proceedings, 2004, pp.145-154.

[16] V. Sadana and X. F. Liu, Analysis of Conflicts among Non-functional Requirements Using Integrated Analysis of Functional and Non-Functional Requirements in Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International, 2007, vol. 1, pp.215-218.

[17] T. L. Heng and L. T. Ming, Using multi-coordinated views with agent communication protocol to detect and resolve inconsistent requirements to improve accuracy in Information Technology (ITSim), 2010 International Symposium in, 2010, vol. 2, pp. 1041-1044.

[18] T. Moser, D. Winkler, M. Heindl, and S. Biffl, Automating the detection of complex semantic conflicts between software requirements in The 23rd International Conference on Software Engineering and Knowledge Engineering, Miami,2011.

[19] M. J. Escalona, M. Urbieta, G. Rossi, J. A. Garcia-Garcia, and E. R. Luna, Detecting Web requirements conflicts and inconsistencies under a model-based perspective, J. Syst. Softw., vol. 86, no. 12, pp. 3024-3038, Dec.2013.

[20] M. Urbieta, M. J. Escalona, E. R. Luna, and G. Rossi, Detecting ConflictsandInconsistenciesinWebApplicationRequirementsinCurrent Trends in Web Engineering, A. Harth and N. Koch, Eds. Springer Berlin Heidelberg, 2012, pp.278-288.

[21] Z. Chentouf, Managing OAM&P requirement conflicts, J. King Saud Univ.-Comput.Inf.Sci.,vol.26,no.3,pp.296-307,Sep.2014.

[22] M.Shehata, L.Jiang, and A. Eberlein, Are requirements interaction detection process guide in Canadian Conference on Electrical and Computer Engineering, 2004, 2004, vol. 3, pp. 1753-1756 Vol.3.

[23] A. Sardinha, R. Chitchyan, J. Arajo, A. Moreira, and A. Rashid, Conflict Identification with EA-Analyzer in Aspect-Oriented Requirements Engineering, A. Moreira, R. Chitchyan, J. Arajo, and A. Rashid, Eds. Springer Berlin Heidelberg, 2013, pp. 209-224.

[24] D. Mairiza, D. Zowghi, and V. Gervasi, Utilizing TOPSIS: A Multi Criteria Decision Analysis Technique for Non-functional Requirements Conflicts in Requirements Engineering, D. Zowghi and Z. Jin, Eds. Springer Berlin Heidelberg, 2014, pp. 31-44.

[25] A. Alebrahim, S. Fabender, M. Heisel, and R. Meis, Problem-Based Requirements Interaction Analysis in Requirements Engineering: Foundation for Software Quality, C. Salinesi and I. van de Weerd, Eds. Springer International Publishing, 2014, pp. 200-215.

[26] A. Egyed and P. Grunbacher, identifying requirements conflicts and cooperation: how quality attributes and automated traceability can help, IEEE Softw., vol. 21, no. 6, pp. 50-58, Nov. 2004.

[27] T. H. Nguyen, B. Q. Vo, M. Lumpe, and J. Grundy,

KBRE: a framework for knowledge-based requirements engineering, Softw. Qual. J., vol. 22, no. 1, pp. 87-119, Apr. 2013.

[28] D. Mairiza and D. Zowghi, Constructing a Catalogue of Conflicts among Non-functional Requirements in Evaluation of Novel Approaches to Software Engineering, L. A. Maciaszek and P. Loucopoulos, Eds. Springer Berlin Heidelberg, 2011, pp. 31-44.

[29] C.-L. Liu, Ontology-Based Requirements Conflicts Analysis in Activity Diagrams in Computational Science and Its Applications – ICCSA,2009.

[30] How to Deal with Stakeholders Conflicts in Requirements Gathering? [Online] Available: https://www.researchgate.net/post/How to deal with stakeholders conflicts in requirements gathering2. [Accessed:20-Apr-2016].

[31] Scope - How Do You Manage Conflicting Stakeholder De- mands? - Project Management Stack Exchange. [Online]. Avail- able: http://pm.stackexchange.com/questions/1399/how-do-you-manage- conflicting-stakeholder-demands. [Accessed:20-Apr-2016].

[32] I Have Two Business Stakeholders Who Have Conflicting Requirements Project connections.com.[Online].http://www.projectconnections.com/knowhow/burning-questions/resolving-conflicting-stakeholder-requirements.html. [Accessed:20-Apr-2016].

[33] Conflicting Requirements.[Online].Available: http://c2.com/cgi/wiki?ConflictingRequirements. [Accessed: 20-Apr- 2016]

[34] K.K. Fletcher, X.F. Liu, M. Tang, Elastic personalized non-functional attribute preference and tradeoff based service selection, ACM Trans. Web 9 (1) (2015) 1–26.

[35] Merilinna, Janne, Anton Yrjönen, and TomiRäty, "NFR+ framework method to support bi-directional traceability of non-functional requirements," Computer Science-Research and Development, pp. 1-15, 2015.

[36] Hu, Haibo, Qi Ma, Tingting Zhang, Yong Tan, Hong Xiang, Chunlei Fu, and Yong Feng,"Semantic modelling and automated reasoning of non-functional requirement conflicts in the context of soft goal interdependencies," IET Software, vol. 9, December 2015, no. 6, pp.145-156, doi:10.1049/iet-sen.2014.0153.

[37] V. Gervasi, D. Zowghi," Reasoning about inconsistencies in natural language requirements", ACM Trans. Softw. Eng. Methodol. 14 (3) (2005) pp.277–330.

[38] Liu, C.L, "CDNFRE: Conflict detector in non-functional requirement evolution based on ontologies," Computer Standards & Interfaces, vol. 47,2016, pp.62-76, doi:10.1016/j.csi.2016.03.002.

[39] . S. A. Mala and G. V. Uma, "Elicitation of non-functional requirements preference for actors of use case from domain model," Lecture Notes in Computer Science, vol.4303/2006, pp. 238-243, 2006.

[40] U.Shah ,S. Patel ,D.Jinwala , "An Ontological Approach to Specify Conflicts among Non-functional requirements", ICGDA, ACM ISBN 978-1-4503-6245-0/19/03, March 2019, DOI:10.1145/3318236.3318257.

[41] Chung, Lawrence, Brian A. Nixon, Eric Yu, and John Mylopoulos, "Non-functional requirements in software engineering," Springer Science & Business Media, vol. 5. 2012.

[42] R.M. Carvalho. ," Dealing with Conflicts between Non-functional Requirements of UbiComp and IoT Applications". In 25th IEEE International Requirements Engineering Conference, RE,2017.

[43] In, H.& Boehm, B., "Using Win-Win quality requirements management tools: a case study", Annals of Software Engineering, vol. 11, p.141-174,2001.

[44] B. Murgante, O. Gervasi, A. Iglesias, D. Taniar, and B. O. Apduhan, Eds. Springer Berlin Heidelberg, 2011, pp. 383-397.

[45] O. Gervasi, D. Taniar, B. Murgante, A. Lagan, Y. Mun, and M. L. Gavrilova, Eds. Springer Berlin Heidelberg, 2009, pp. 1-12.

[46] Wiegers.K., Beatty.J, "software requirements, third edition", Microsoft Press ,Redmont , Washington,2013,pp.287-290.

[47] Shah U, Patel S, Jinwala DC. Detecting Intra-Conflicts in Non-functional Requirements. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems. 2021 Jun;29(03):435-61.

[48] Zhang X, Wang X. Tradeoff analysis for conflicting software non-functional requirements. IEEE Access. 2019 Oct 23; 7:156463-75.

[49] Afraz S, Rashidi H, Mikaeilvand N. CSOP+ RP: a novel constraints satisfaction model for requirements prioritization in large-scale software systems. Turkish Journal of Electrical Engineering and Computer Sciences. 2021;29(7):3074-91.

[50] Benesty J, Chen J, Huang Y, Cohen I. Pearson correlation coefficient. In Noise Reduction in Speech Processing. Springer; 2009:1- 4.

[51] Myers JL, Well A, Lorch RF. Research Design and Statistical Analysis. Routledge; 2010.

[52] Kendall MG. A new measure of rank correlation. Biometrika. 1938;30(1/2):81.