International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:4, No:8, 2010

Conflicts Identification among Non-functional Requirements using Matrix Maps

Abdul H, Jamil A, and Imran U

***Abstract*—**Conflicts identification among non-functional requirements is often identified intuitively which impairs conflict analysis practices. This paper proposes a new model to identify conflicts among non-functional requirements. The proposed model uses the matrix mechanism to identify the quality based conflicts among non-functional requirements. The potential conflicts are identified through the mapping of low level conflicting quality attributes to low level functionalities using the matrices. The proposed model achieves the identification of conflicts among product and process requirements, identifies false conflicts, decreases the documentation overhead, and maintains transparency of identified conflicts. The attributes are not concomitantly taken into account by current models in practice.

***Keywords*—**Conflict Identification, Matrix Maps, Non-functional Requirements, Requirements Analysis, Software Engineering

I. Introduction

BESIDES implementing all the desired functionality, it is highly desirable for software systems to cope with non-functional aspects as well. These may include reliability, security, accuracy, safety, performance, look and feel requirements, as well as organizational, cultural, and political requirements. These non-functional aspects must be treated as nonfunctional requirements (NFRs) of the software [1-2]. Unproductive dealing with NFRs has led to a succession of failures in software development [3-4], including the very mighty reconnoiter of the London Ambulance System [5], where the deactivation of the software right after its deployment was strongly prejudiced by NFRs noncompliance. These requirements have been pointed out in literature [2],[6-11] as the most expensive and complicated ones to deal with.

In spite their importance, NFRs have unexpectedly received little consideration in the literature and are poorly tacit compared to less significant aspects of the software development [2]. The majority of the work on NFRs uses a product-oriented approach, which is concerned with measuring how often a software system is in harmony with the set of non-functional requirements that it should satisfy [10-14].

A. Hannan is working as Research Fellow/Lecturer in IQRA University Islamabad Campus, Pakistan (phone: +92-331-623-1304; fax: +92-51-; e-mail: hannan@iqraisb.edu.pk).

J. Ahmed was with Ghulam Ishaq Khan Institute (G.I.K.I.) as Head of Department of Computer Science. He is now Dean of IQRA University Islamabad Campus, Pakistan (e-mail: jamil@iqraisb.edu.pk).

I. Usman is with the Computing and Technology Department, IQRA University Islamabad Campus, Pakistan, as an Asst. Professor (e-mail: imran.usman@gmail.com).

Non-functional requirements include constraint and quality. Quality attributes are properties of the system that its stakeholders care about, and consequently, will affect their level of contentment with the system. Constraints are the scope of the quality. Since constraints are not under consideration for the negotiation process and, unlike qualities, are theoretically exclusive during design trade-offs [13].

One of the critical areas in software engineering is the requirements conflict identification. Since non-functional requirements are under discussion, conflicts often arise when two quality-attributes have an opposite behavior to each other. There is actually a significant difference between the specific requirements of the same task. Numerous software projects have failed because they contained impoverished set of non-functional requirements, even though, they certainly may have had a good set of functional and interface requirements [14]. The primary motivation for this particular research is an indispensable step towards achieving successful software requirements in order to achieve the right balance of non-functional requirements. To achieve this task, many requirement engineering techniques are necessary and important. That is why a remarkable number of leaders in the area believe that a relatively more powerful technique is the *requirement conflict identification* rather than *requirement conflict negotiation*.

There are number of models and techniques [14-17] which identify the conflicts among the non-functional requirements based on quality attributes. The problem with quality based conflict identification is that the false conflicts are also identified along with the potential conflicts. False conflicts are the contradictions among the non-functional requirements which are, in fact, not conflicts but are identified under the umbrella of a particular quality attribute. Hence, it causes an overhead in the conflict negotiation process.

The rest of the work is organized as follows. Section 2 discusses some related work in literature by contemporary researchers. Section 3 discusses the proposed model. Section 4 presents results and discussion. Section 5 discusses the conclusion and section 6 provides the references.

II. Related work

A variety of conflict identification models are proposed in literature. These include Sandana and Lui [15] model based on analysis and detection of conflicts among non-functional requirements using integrated analysis of functional and non-functional requirements "ACONIAN". This framework is based on non-functional decomposition "N.F.D." proposed by Poort and deWith [18]. Poort provided a model to transform the conflicting requirements into a system.

Bertagnolli and Lisboa [17] proposed a model which deals with the requirements from requirements phase to design phase using aspect oriented software engineering. The conflict identification methodology is based on the set theory. Another approach is proposed by Egyed and Grunbacher [16]. It comprises of identification of conflicts and co-operations with the help of quality attributes and automated traceability. Boehm and In [19] used a knowledge based tool, Software Cost Option Strategy Tool SCOST, which deals with the conflicts among the process requirements. It assists the stakeholders to surface and negotiate conflict and risks among the requirements.

In, Kim, Yun and Yau [20] proposed a quality of service conflict identification model for situation aware middleware. Since the application need middleware and it changes as the application change, so In proposed a quality of service resource conflict identification model which analyzes whether the quality of service requirements are met and what tradeoff relationship are present among requirements.

The experimental study of the above mentioned approaches concludes that most of the current approaches work in an ad hoc manner by identifying conflicts either for product requirements or process requirements. Secondly, these approaches identify conflicts based on the quality attributes which result the presence of false conflicts along with the potential conflicts. Lastly, the aforementioned approaches are difficult to implement in form of paper implementation. Therefore, we propose a new model in this work which identifies conflicts among the non-functional requirements using matrices by mapping low level conflicting quality attributes to low level functionalities. The following section elaborates our proposed approach and thus, devises the proposed model.

III. The Proposed Model

The proposed model is an extension of Sandana and Lui's [15] work. It not only identifies the conflicts among non-functional requirements but also analyzes them by mapping low level conflicting quality attributes over low level functionalities using matrices. The following steps achieve the task.

step 1. Write high level non-functional requirements in structured form and identify quality and functionality attributes in each non-functional requirement.
step 2. Make the hierarchy of functionality attribute and identify low level functionalities.
step 3. Make the hierarchy of quality attribute based on the major functionalities.
step 4. Identify conflicts using quality-to-quality matrix.
step 5. Identify potential conflicts by mapping low level conflicting low level quality attributes over low level functionalities.

Fig. 1 shows our proposed model. It starts with structured representation of high level non-functional requirements. Conflict identification is then performed on the high level quality attribute in order to identify the low level conflicting quality attributes. By doing so, low level quality attributes are under consideration. These low level conflicting quality attributes are then mapped to low level functionalities in order to identify potential conflicts.

The structured representation for requirement statements, adopted for this model, is a modified representation presented in [18], [15]. The constraints have been removed from the structured representation based on [9] which says that constraints are not issue in negotiation. The modified representation can be presented either of the following two structured forms.

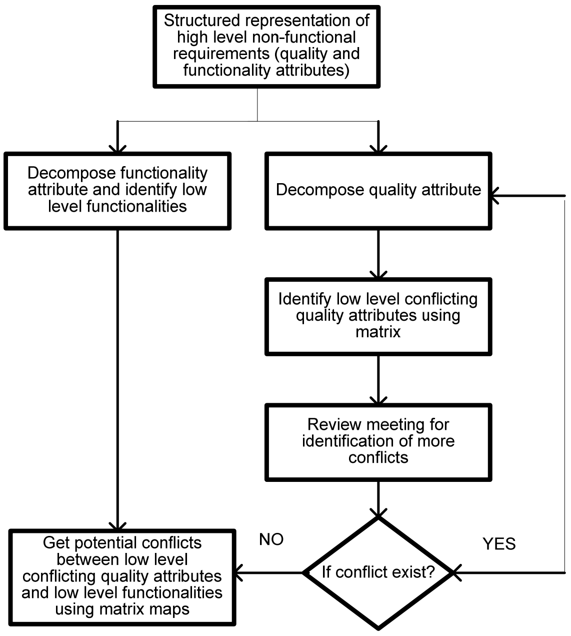

Structured representation of high level non-functional requirements (quality and functionality attributes)
Decompose functionality attribute and identify low level functionalities
Decompose quality attribute
Identify low level conflicting quality attributes using matrix
Review meeting for identification of more conflicts
If conflict exist?
NO
YES
Get potential conflicts between low level conflicting quality attributes and low level functionalities using matrix maps

Fig. 1 Model for analysis and detection of conflicts among non-functional requirements

SF 1: Quality Attribute [Subject S] of Functionality should be [Verb V] constraint.

SF 2: Functionality should have [Verb V] constraint Quality Attribute [Subject S].

Where S represents the quality attribute and V represents the functionality.

The proposed model uses matrices to analyze and detect the conflicts among the non-functional requirements. These matrices are used as tool for the identification of conflicts among the non-functional requirements. In this research, we have considered same example of search engine as mentioned in [15]. For clarity purpose, the following hierarchy diagram and table are reproduced from [15]. However, we only deal with low level functionality attributes and quality attributes, therefore, the hierarchy diagrams of functionality attribute and quality attribute in [15] are modified and presented in Fig. 2 and Fig. 3 respectively. The high level functionality of a search engine is "Search" and the high level quality attribute is "Quality". The example starts with the following high level non-functional requirement in structured form: "*The quality of search should be high*"

The high level functionality "Search" and high level quality attribute "Quality" are decomposed. The "Quality" is decomposed on the basis of major low level functionalities such as "Search by Title", "Search by keyword", "Use Boolean Logic" and "Use Case (in) Sensitivity Options".

After the decomposition of high level functionality and quality attribute, the conflicts are identified on the basis of quality-to-quality matrix as shown in Table I.

In this table, **O** shows the supporting attributes and **X** shows the conflicting attributes. From this table, the low level conflicting quality attributes are identified and separated.

Now the low level quality based non-functional requirements in Table II are mapped to low level functionalities identified during the hierarchy process of high level functionality. The potential conflicts are identified through the matrix shown in Table III.

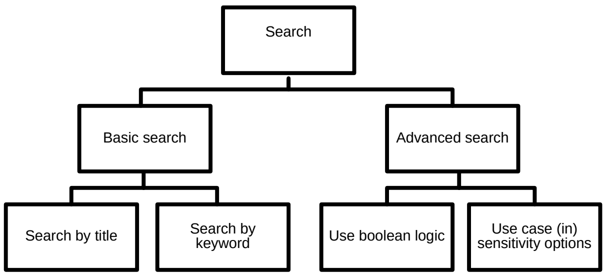


Fig. 2 Hierarchy of high level functionality "Search"

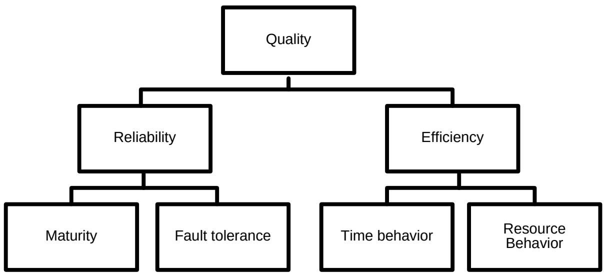


Fig. 3 Hierarchy of high level quality attribute "Quality"

With reference to Table II, "Maturity" and "Fault tolerance" both have the conflict with same quality attribute i.e. "Time behavior". So in Table III, "Time behavior" is shown in a single column for simplicity. In Table III, the conflict between the quality attributes "Time Behavior" and "Fault Tolerance" with respect to low level functionalities does not exist.

TABLE I
CONFLICT IDENTIFICATION USING QUALITY-TO-QUALITY MATRIX

Quality Attributes	Maturity	Fault Tolerance	Time Behavior	Resource Behavior
Maturity	O	O	X	O
Fault Tolerance	O	O	X	O
Time Behavior	X	X	O	O
Resource Behavior	O	O	O	O

TABLE II
QUALITY BASED CONFLICTING AND NON-CONFLICTING NFRs

High level NFR	Low level NFRs	Low level quality based conflicting NFRs	Non-conflicting NFRs
Quality of search should be high	1. Maturity of search should be high. 2. Fault tolerance of search should be high. 3. Time behavior of search should be high. 4. Resource behavior of search should be high	1. "Maturity of search should be high." *versus* "Time behavior of search should be high." 2. "Fault tolerance of search should be high." *versus* "Time behavior of search should be high."	1. Resource behavior of search should be high.

TABLE III
QUALITY-TO-FUNCTIONALITY MATRIX

Functionality/Quality	Maturity	Time behavior		Fault tolerance
Search by title	X	X	O	O
Search by keyword	X	X	O	O
Use Boolean logic	X	X	O	O
Use case (in) sensitivity options	O	O	O	O

Whereas, “Maturity” and “Time Behavior” are conflicting in nature with respect to “Search by Title”, “Search by Keyword” and “Use Boolean Logic”. The potential conflicts are mentioned in Table IV.

In Table IV, the quality based conflicting non-functional requirements (extracted from Table II) are further elaborated on the bases of low level functionality and conflicts are classified into potential conflicts and non-conflicting non-functional requirements.

TABLE IV
IDENTIFICATION OF POTENTIAL CONFLICTS ALONG WITH NON CONFLICTING NFRs

Quality based conflicting NFRs	Functionality mapped low level NFRs	NFRs with potential conflicts	Non-conflicting NFRs
Maturity of search should be high.	1. Maturity of search by title should be high. 2. Maturity of search by keyword should be high. 3. Maturity of use Boolean logic should be high. 4. Maturity of use case (in) sensitivity should be high.	1. Maturity of search by title should be high. *versus* Time behavior of search by title should be high. 2. Maturity of search by keyword should be high. *versus* Time behavior of search by keyword should be high. 3. Maturity of use Boolean logic should be high. *versus* Time behavior of use Boolean logic should be high.	1. Time behavior of search by title should be high. 2. Fault tolerance of search by title should be high. 3. Time behavior of search by keyword should be high. 4. Fault tolerance of search by keyword should be high. 5. Time behavior of use Boolean logic should be high. 6. Fault tolerance of use Boolean logic should be high. 7. Time behavior of use case (in) sensitivity should be high. 8. Fault tolerance of use case (in) sensitivity should be high. 9. Maturity of use case (in) sensitivity should be high. 10. Time behavior of use case (in) sensitivity should be high. 11. Resource behavior of search by title should be high. 12. Resource behavior of search by keyword should be high. 13. Resource behavior of use Boolean logic should be high. 14. Resource behavior of use case (in) sensitivity should be high.
Time behavior of search should be high	1. Time behavior of search by title should be high. 2. Time behavior of search by keyword should be high. 3. Time behavior of use Boolean logic should be high. 4. Time behavior of use case (in) sensitivity should be high.		
Fault tolerance of search should be high.	1. Fault tolerance of search by title should be high. 2. Fault tolerance of search by keyword should be high. 3. Fault tolerance of use Boolean logic should be high. 4. Fault tolerance of use case (in) sensitivity should be high.		
Time behavior of search should be high.	1. Time behavior of search by title should be high. 2. Time behavior of search by keyword should be high. 3. Time behavior of use Boolean logic should be high. 4. Time behavior of use case (in) sensitivity should be high.		

In Table II, the quality attributes *Maturity* and *Fault tolerance* are conflicting with *time behavior* of search. These quality attributes are mapped over low level functionalities of *search*. Since *search* is a high level functionality, the low level functionalities are identified in Fig. 2. Table III shows that the quality attributes *maturity* and *time behavior* are potential conflicts based on low level functionalities *search by title*, *search by keyword* and *use Boolean logic*. The *fault tolerance* and *time behavior* don't have any conflict on the basis of low level functionalities.

IV. RESULTS AND DISCUSSION

Following are the major achievements of this research which are actually based on the identified problems mentioned in section 2.

Product and process requirements conflicts identification:

The proposed model not only identifies the potential conflicts among the product requirements but also it identifies the conflicts among the process based requirements

False conflict identification:

Due to the fact that quality based conflict identification has an inherited problem of the false conflicts, the proposed model identifies the false conflicts through the mapped analysis of low level functionality over conflicting low level quality attributes through a matrix.

Simple implementation

The proposed model takes very less overhead of documentation as compared to [15] because it reduces number of diagrams and tables.

Transparency of conflicts:

The results obtained from this proposed model are very transparent. We do not need to back track to identify the conflicting requirements as we do in [15].

Most of the models for conflict identification in literature deal with the product requirements. They simply skip the impact of the process requirements on the product requirements or vice versa. Table V shows that all the models deal with the product requirements but only the "ACONIAN" and the proposed model also deals with the process requirements. The quality based conflict identification results in the origination of the false conflicts which are the overhead to the conflict negotiation process. Only the proposed model and the "identification of conflicts using integrated analysis of functional and non-functional requirements" identify the false conflicts.

TABLE V
COMPARISON TABLE

Conflict models	Product req.	Process req.	False conflicts identification	Document overhead	Transparent conflicts
ACONIAN	Yes	Yes	Yes	Yes	No
FRIDA model	Yes	No	No	No	No
Conflict & co-operation	Yes	No	Yes	Yes	No
Proposed model	Yes	Yes	Yes	No	Yes

The objective of this work is that the paper implementation should be simple. The proposed model and FRIDA model shows the ideal behavior. In case of the transparency of conflicts, only proposed model shows transparent conflict identification. The rest of the models are either does not identify the false conflicts or the back tracking is involved to get the transparent conflicts. So in the comparison Table V, the proposed model shows the ideal behavior for each comparison parameter.

V. CONCLUSION

We propose a model which not only represents a modified semantics of NFRs analyze and detect conflicts among non-functional requirements using the matrices to map low level conflicting quality attributes over the low level functional attributes. The model enables us to capture the potential conflicts based on the relationship among quality attributes and functionalities. The use of mapping the low level conflicting quality attributes over low level functionalities is very promising. Since the constraints are not the area of interest in negotiation, so there is no need of constraints in the conflict identification process. The constraints are not involved because constraints are the scope of the quality attributes and they can't be decomposed. If a constraint is able to decompose then another high level non-functional requirement must be generated in which the decomposable constraint should be a quality attribute with some functionality.

ACKNOWLEDGMENT

We want to thank the management of IQRA University Islamabad Campus for providing a research oriented and research friendly environment to the faculty as well as the students.

Special thanks to Mr. Muhammad Shoaib, Mr. Muhammad Aamir and Mr. Mehmood Ahmad for their critical and valuable discussion.

REFERENCES

[1]. L. Chung and B. Nixon, "Dealing with Nonfunctional Requirements: Three Experimental Studies of a Process-Oriented Approach," Proc. 17th Int'l Conf. Software Eng., Apr. 2000, pp. 24-28.

[2]. L. Chung, B. Nixon, E. Yu, and J. Mylopoulos, "Non-Functional Requirements in Software Engineering", Kluwer Academic Publisher, London, 1999.

[3]. K. K. Breitman, J. C. S. P. Leite, and A. Finkelstein, "The World's Stage: A Survey on Requirements Engineering Using a Real-Life Case Study," The Brazilian Computer Soc., July 1999, vol. 6, no. 1, pp. 13-38.

[4]. D. R. Lindstrom, "Five Ways to Destroy a Development Project," IEEE Software, Sept. 1993, vol. 10, no. 5, pp. 55-58.

[5]. A. Finkelstein and J. Dowell, "A Comedy of Errors: The London Ambulance Service Case Study," Proc. Eighth Int'l Workshop Software Specification and Design, 1996, pp. 2-5.

[6]. F. P. Brooks Jr., "No Silver Bullet: Essences and Accidents of Software Engineering", IEEE Computer, Apr. 1987,vol. 20, no. 4, pp. 10-19.

[7]. L. M. Cysneiros and J. C. S. P. Leite, "Integrating Non-Functional Requirements into Data Modeling," Proc. Fourth Int'l Symp. Requirements Engineering Ireland, , June 1999, pp. 162-171

[8]. A. Davis, "Software Requirements: Objects, Functions and States." Prentice Hall, ed. 2, 1993.

[9]. R. Malan, D. Bredemeyer, "Defining the non-functional requirements", Bredemeyer Consulting, Bloomington, 2001

[10]. N. A. Ernst, Y. Yijun and J. Mylopoulos "Visualizing non-functional requirements" First Int. workshop on Requirements Engineering Visualization REV' 06, Sept. 2006, pp.2,

[11]. P. Morris, "Counting non-functional requirements when they are implemented as software", Total Metrics, version 2.2, 2006

[12]. J. Musa, A. Lannino, and K. Okumoto, "Software Reliability: Measurement, Prediction and Application." McGraw-Hill Inc., New York, 1989.

[13]. N. E. Fenton and S. L. Pfleeger, "Software Metrics: A Rigorous and Practical Approach", International Thomson Computer Press, ed. 2, 1996

[14]. H. In "Conflict Identification and Resolution for Software Attribute Requirements", PhD Thesis, Faculty of Graduate School, University of Southern California, 1998

[15]. V. Sandana and X. F. Liu, "Analysis of Conflicts Among Non-functional Requirements using Integrated Analysis of Functional and Non-functional requirements" 31st IEEE Annual International Computer Software and Applications Conference, Beijing, 2007

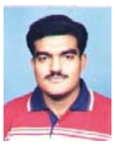
[16]. A. Egyed and P. Grunbacher "Identifying Requirements Conflicts and Cooperation: How Quality Attributes and Automated Traceability Can Help", IEEE Software, , 2004, vol. 21, no. 6, pp. 50-58

[17]. S. C. Bertagnolli and M. L. B. Lisboa "The FRIDA Model" 2000

[18]. E. R. Poort and P. H. N. deWith, "Resolving Requirement Conflicts through Non-Functional Decomposition" Proc. Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA'04), IEEE Computer Society, 2004

[19]. B. Boehm and H. In, "Software Cost Option Strategy Tool (S-COST)" COMPSAC96, Seoul, Korea: IEEE Comp. Society Press, 1996

[20]. H. In, C. H. Kim, U. Yun and S. S. Yau, "Q-MAR: A QoS resource Conflict Identification Model for Situation-Aware Middleware" Proc. 9th IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS'03), IEEE Computer Society, 2003



Abdul H currently working as a Lecturer in IQRA University Islamabad Campus, Pakistan. He earned his bachelor degree from Foundation University Islamabad, Islamabad, Pakistan with majors in software engineering. He did his MS from IQRA University Islamabad campus, Pakistan with majors in software engineering.

He is also working as a research associate in IQRA University-Computational intelligence Research Group (IU-CIRG). Currently his interest area is intelligent software engineering.

Jamil A is Dean IQRA University Islamabad Campus, Pakistan. He is Group Leader of IQRA University-Computational Intelligence Research Group. His interest areas are artificial intelligence and neural networks.

Imran U is working as Asst. Prof. in IQRA University Islamabad Campus, Pakistan. He is the member of IQRA University-Computational Intelligence Research Group (IU-CIRG). His interest area is digital water marking.