

Verification and validation of IM-DeCRuD approach using DESMET for its applicability

Jamaluddin Jasmis¹, Shamsul Jamel Elias², Azlan Abdul Aziz³, Mohd Zaki Zakaria⁴
R Badlishah Ahmad⁵, Roshidi Din⁶, Rosmadi Bakar⁷

^{1,3}Faculty of Computer and Mathematical Sciences, UiTM (Melaka) Jasin Campus, Malaysia

²Universiti Teknologi MARA (UiTM) Kedah, Malaysia

⁴Universiti Teknologi MARA (UiTM) Shah Alam, Malaysia

⁵Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin (UniSZA), Malaysia

^{6,7}School of Computing, College of Arts and Sciences, UUM, Malaysia

Article Info

Article history:

Received Aug 15, 2018

Revised Dec 7, 2018

Accepted Dec 19, 2018

Keywords:

DESMET

IM-DeCRuD

Requirements crosscutting

V&V

ABSTRACT

Requirements crosscutting in software development and maintenance has gradually become an important issue in software engineering with a growing need of traceability support to better understand the requirements crosscutting processes to comply with industrial standards. However, many recent works focusing on identification, modularization, composition and conflict dissolution of requirements crosscutting are mostly saturated at requirements level. Hence, these works fail to specify crosscutting properties for functional and non-functional requirements at other phases leading to insufficient support for software engineers. Recently, a new approach called the Identification, Modularization, Design Composition Rules and Conflict Dissolutions (IM-DeCRuD) was proposed to provide a special traceability to facilitate better understanding and reasoning towards requirements crosscutting implementation, along with a tool as a proof-of-concept. In this paper, the tool was evaluated and the results were verified by some experts in the industries. The feedbacks on applicability aspect were then gathered and analyzed using DESMET qualitative method. The outcome showed that the IM-DeCRuD was applicable to cope with the tedious engineering processes in handling crosscutting properties at requirements, analysis and design phases for system development and evolution.

Copyright © 2019 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Jamaluddin Jasmis,
Faculty of Computer and Mathematical Sciences,
UiTM (Melaka) Jasin Campus,
77300 Merlimau, Melaka, Malaysia.
Email: jamaluddinjasmis@salam.uitm.edu.my

1. INTRODUCTION

Requirements is defined as “any matters of interest in a software system” which can be related to system functionalities or its properties [1]. It can be classified into functional (system’s behavior or subsystems) and non-functional (system’s properties). A requirement rarely acts standalone as most may influence or constraint other requirements. This type of scenario is called crosscutting [2]. Crosscutting is usually described in terms of scattering and tangling. For example, a stakeholder’s functional requirement with a capability of handling a user on-line transaction might be described by some properties i.e. the non-functional requirements such as user’s response time within acceptable limit, with appropriate security features and affordable workload. This type of scenario is called tangling. In another situation, a non-functional requirement may describe properties for several other functional requirements in order for the functional requirements to remain

useful. For example, the performance as a property of a system would be applied to several other functional requirements i.e. concerns with similar or different specifications. This type of scenario is called scattering.

Requirements crosscutting are related to each other within artifact as well as correlated artifacts across multiple phases [3-5]. Consequently, any changes to requirements crosscutting may yield direct or indirect impact to other artifacts. Most of recent works are focusing on identification, modularization, composition and conflict analysis of requirements crosscutting solely at requirements level. It is due to its straightforwardness in dealing with high-level language in requirements documentations to specify requirements [6]. However, there is significant research gap to appropriately specify crosscutting properties for functional and non-functional concerns at both requirements and design phases. Due to this absence, software engineers have no appropriate guidelines to attend to crosscutting concerns across development stages [6, 7].

Several state-of-the-art approaches had been evaluated using the criteria obtained from the literature. The results showed us that so far, there is no single approach that fully satisfied with all the capabilities that have to be fulfilled to support requirements crosscutting [8]. Apparently, we proposed a new approach called the Identification, Modularization, Design Composition Rules and Conflict Dissolutions (IM-DeCRuD) has been done that provides a special traceability to facilitate better understanding and reasoning for engineering tasks towards requirements crosscutting during software development and evolution. Moreover, it also promotes a simple but significant way to support pragmatic changes of crosscutting properties at requirements, analysis and design phases for medium sizes of software development and maintenance projects [9, 10]. A tool has been developed based on the proposed approach to store relationship dependencies since traceability is highly considered among artifacts in Model Driven Engineering (MDE) to support understanding and maintenance of software systems [3, 11].

In this paper, we aim to evaluate the results that were generated by applying IM-DeCRuD approach to *myPolicy* case study by the domain experts to ensure its applicability. The evaluation method for this research was based on the evaluation guidelines proposed in DESMET [12]. DESMET is a project that attempts to provide practical evaluation approach in software engineering.

Organizations of this paper are as follows: Section 2 presents the implementation of DESMET and Section 3 discusses the overall results. Finally, Section 4 presents the conclusion.

2. RESEARCH METHOD

DESMET is chosen as the evaluation methodology as it is mainly used to evaluate the software engineering methods and tools. There are nine manners proposed by DESMET [13] including quantitative and qualitative manners. Since applicability can be measured by determining the appropriateness of features available in the proposed approach, feature analysis is selected as the purpose of the qualitative method in this research. This analysis might be formally organized in qualitative case study and survey. These settings suggest one person or a group of potential experts to undertake the evaluation [14]. Hence, a group of four working people with some working experiences were employed for this evaluation. Three steps are proposed by DESMET: *identifying features*, *scoring features* and *analysis*. These steps are briefly explained as following:

a) Identifying Features

Features are of two types; *simple* and *compound* features. The features that need to be assessed are *concerns identification*, *composition*, *conflicts identification* and *traceability support*; following would be *mapping*, *maintenance* and *GUI support*.

b) Scoring Features

Simple features are evaluated by YES or NO whereas compound features shall be measured on an ordinal scale [13]. However, each compound feature should be along with an assessment of its importance and conformance. The scales for measuring importance and conformance will be discussed below:

a. Importance

The importance of a feature can be judged by two ways; by examining whether it is mandatory or only desirable. This opinion of importance leads to two evaluation criteria. First analyze whether a feature is mandatory, and the second assesses the degree to which a non-mandatory feature is desirable. To assess a feature, following four scaling points are expressed: *Mandatory*, *Highly Desirable*, *Desirable* and *Nice to Have*.

b. Conformance

The assessment scale for conformance has two objectives; defining the level of support required for an individual feature and providing the evaluator with a measurement scale that will score the feature of a particular candidate. The granularity of the scale points is dependent upon the features that are to be considered. It may be from 5 (full support) to 0 (no support).

c) Analysis

Feature analysis gives clear picture of a tool whether it fulfills the demands of the potential users or not. After providing the importance and conformance of features, the score sheets must be analyzed.

Predicated on the DESMET method, if the acceptance threshold is elaborated, the analysis must base on the difference between the acceptance threshold set by the users for every feature and the score gained for that feature. If the acceptance threshold is not achievable, the assessment should be predicated on the scores of the approaches relative to one another. The latter approach has been used as the acceptance threshold is not achievable in this research. Therefore, the analysis must be based on accumulating the absolute scores.

The combined score for one feature set would be the sum of the conformance values of all features for a certain approach, which may be expressed as a percentage of the maximum score. For example, suppose that the combined score for three features is 11 out of 15 (the maximum score), then the converted percentage score would be 73%. Finally, the overall score can be obtained by determining the aggregate score for each feature set.

2.1. Subjects and Environment

Several sessions for this evaluation were conducted where these subjects (domain experts) were selected whom background are ranged from IT manager as well as freelance software developer that involves in many software development projects from government agencies. From investigation, all the experts had some working experiences for more than six years in the software industries. During the session, IM-DeCRuD tool was demonstrated and results of the case study were briefed.

2.2. Questionnaires

The questionnaire was designed to accommodate the domain experts' evaluations on IM-DeCRuD approach. The main objective of this evaluation was to assess the applicability of the identification and modularization, design composition rule and conflict analysis for crosscutting concerns. The questionnaire consists of two sections (see Appendix C); Section A and Section B. Section A was designed to relate to the previous professional background of the subjects and Section B was used to get the scores for the features and overall usefulness of the prototype tool. There were nine questions related to the features of IM-DeCRuD. Participants were invited to provide some degree of its usefulness with respect to identification and modularization, design composition rule, conflict analysis, maintainability, scalability and GUI. The experts were also asked to give some comments on the general performance of IM-DeCRuD as an open question.

The independent variables used with the IM-DeCRuD prototype were, the subjects and the myPolicy case study, while the dependent variables were the scores of questions asked. The scheme of score evaluation was formulated based on [15].

2.3. Evaluation Procedures

Four subjects participated in two separate sessions where each session occupied approximately two hours with the above mentioned myPolicy case study. Subjects were given briefing sessions on the underlying concept of crosscutting concerns, the approach and the prototype tool before the case study being demonstrated. They were also provided with a set of questionnaires which require them to select one answer from available options.

2.4. Possible Threats and Validity

Following factors may kick in to some possible threats in this evaluation such as:

- a) DESMET proposes two kinds of evaluation methods, namely as quantitative and qualitative. The later was adapted due to IM-DeCRuD underlies specific principles and user population for this kind of approach is not generic [13]. Due to this situation, as such, this evaluation does not encompass any benchmarking in term of comparative study on statistical results of IM-DeCRuD and other approaches or tools, applied onto the same case study (myPolicy) in which the researcher found it is hard to conduct due to limited number of subjects, human commitment and time. Moreover, the objective for this evaluation is to ensure IM-DeCRuD's practicability towards the industrial-strength case study.
- b) It can be issue of unfamiliarity with the tool/approach for evaluation. To overcome this issue, a short briefing was arranged on the utilization of IM-DeCRuD prior to its evaluation. The sessions were conducted on all subjects under proper supervision. A total of six hours on three sessions were allocated in order to secure a good result and to keep their enthusiasm during the evaluation time. It was foreseen that the subjects might very frequently involve in questions and answers during the sessions in order to reach for some acceptable understanding and awareness in this research. To reduce this problem, some useful information had been made available that included documentation in both hard and soft copies for easy references.
- c) The evaluation is based on one person's experience of using the method/tool and the evaluation criteria are subjective. To evade this phenomenon, a group of subjects was selected in order to leverage the evaluation results obtained from them. Moreover, before the experiment, the subjects were analyzed with questions

referring to years of experience and types of projects they have been involved with in software development. The subjects were also given an explanation regarding the skill level to make sure they understand the term used in the skill level (beginner, advanced beginners, competent, proficient and expert). Besides, prior investigations on their level of familiarities and awareness with the underlying specific principles of crosscutting concerns or at least non-functional requirements element was also been done. This was intended to ensure that all subjects will have common understanding as possible.

3. EVALUATION RESULTS

This section discusses the qualitative results based on the tool/method evaluation. Here, the scored results are evaluated.

3.1. Features Evaluation by Users

In this section, results from myPolicy case study were analyzed. The analysis of results is based on the DESMET method as described previously. Furthermore, the type of analysis is *feature analysis*, as suggested by [12]. Based on the DESMET method, there are three steps for feature analysis: *feature identification, feature scoring and analysis*. These steps will be described in the following:

a) Features Identification

The features for assessment of the approach are as follows:

1. *Concerns Identification Support (CI)*: Does the approach support identification of crosscutting concerns and other requirements components?
2. *Composition Support (CS)*: Does the approach accommodate inter-relationship between crosscutting concerns and other requirements components?
3. *Conflict Identification Support (CTI)*: Does the approach provide conflict identification support for conflicting crosscutting concerns?
4. *Traceability Support (TS)*: Does the approach provide traceability for requirements components between requirements and design development phases?
5. *Mapping Support (MP)*: Does the approach support crosscutting concerns mapping to upper level class diagram?
6. *GUI*: Does the approach facilitate graphical user interface?
7. *Maintenance Support (MT)*: Does the approach support software maintenance?
8. *Scalability Support (SC)*: Does the approach support small and big scale software projects?

b) Features Scoring

There are two types of features: simple features and compound features. Simple features are evaluated by answering “YES” or “NO” whereas compound features should be measured on an ordinal scale. Degree of support for compound features is offered by the approach. In this research, GUI is known as simple feature, whereas the remaining are compound features.

Every simple feature must go along with by an assessment of the level of importance. Each compound feature should be associated by an evaluation of their importance and conformance to a specific feature or characteristic. Scales to measure importance and conformance will be described in the following:

Importance

A great technique is one which contains the attributes that are crucial for customers. The importance could be evaluated by seeing whether it is mandatory or desirable. This viewpoint of importance brings about two assessment standards; (i) that indicates whether the feature is obligatory and (ii) that evaluates the degree to which a non-mandatory feature is needed. The following scale points need to be considered to evaluate a feature:

- a) *M* - Mandatory
- b) *HD* - Highly desirable
- c) *D* - Desirable
- d) *N* - Nice to have

Importance of features in this research is shown in Table 1:

Table 1. Importance of Features

Features	CI	CS	CTI	TS	MP	GUI	MT	SC
Importance	HD	HD	M	HD	HD	HD	HD	HD

Moreover, the importance assessment can be considered as a weighting factor. The following weights are suggested by [12] as shown in Table 2.

Table 2. Features Weights

Features	Weight
Mandatory	10
Highly Desirable	6
Desirable	3
Nice To Have	1

Conformance

The objective of the assessment scale for conformance is to define the necessary support level of a specific feature. Additionally it supplies the assessor with a continuous determining scale versus which to rack up the feature of a specific candidate. Following Table 3 presents the scores proposed by [13]. The assessment table for IM-DeCRuD with respect to the above assessment scale is shown in Table 4.

Table 3. Assessment Scale for Features to Support Tool

Generic scale point	Definition of scale point	Scale point mapping
Makes things worse	Cause confusion. The way the feature is implemented makes it difficult to use and/or encouraged incorrect use of the feature.	-1
No support	Fails to recognise it. The feature is not supported nor referred to in the user manual.	0
Little support	The feature is supported indirectly, for example by the use of other tool features in non-standard combinations.	1
Some support	The feature appears explicitly in the feature list of the tools and user manual. However, some aspects of feature use are not catered for.	2
Strong support	The feature appears explicitly in the feature list of the tools and user manual. All aspects of the feature are covered but use the feature depends on the expertise of the user.	3
Very strong support	The feature appears explicitly in the feature list of the tools and user manual. All aspects of the feature are covered and the tool provides tailored dialogue boxes to assist the user	4
Full support	The feature appears explicitly in the feature list of the tools and user manual. All aspects of the feature are covered and the tool provides user scenarios to assist the user such as “Wizards”	5

Table 4. Assessment Table for IM-DeCRuD

Features	CI	CS	CTI	TS	MP	GUI	MT	SC
IM-DeCRuD	4	4	5	4	4	YES	4	4

The scores for all the above mentioned features are described in Table 4. The results in Table 4 show that all the features have been accepted to be highly satisfactory. *Conflict Identification* feature (which is one of the basic themes of this research) obtained the highest average score in which it shows the acceptance of this tool.

From the perspective of overall usefulness, 50% of the subjects chose 5 as Very Useful and another 50% chose 4 as Useful. None of the subjects chose moderate, useless or very useless. It again shows that overall performance of the IM-DeCRuD is considered as excellent.

Now the discussion will be on the individual features of IM-DeCRuD. On the support of Concerns Identification and Composition, 75% of the subjects agree to regard it as Useful and the other 25% chose Very Useful. These results seem to be very promising. With regard to support of Conflict Identification, 75% choose Very Useful, and 25% choose Useful. In terms of Traceability support, all subjects equally chose Useful with 50% scores and Moderate with similar scores as well.

Identical results have been obtained on the support of Mapping and Maintainability where subjects opted all to be Very Useful (25%), Useful (50%) and Moderate (25%). Similar results also have been obtained for GUI and Scalability where the subjects scored both as Useful (75%) instead of Moderate (25%). The best thing is that was no single subject choosing Useless or Very Useless for any feature of the prototype tool.

3.2. Findings of the Analysis

In this section, the findings of the research will be predicated in qualitative perspective based on the study of domain expert evaluations on IM-DeCRuD approach and tool. The prototype tool, IM-DeCRuD,

which is based on this research, was exposed to the subjects by letting them use and evaluate its practicability. Feedback and comments from users regarding its usefulness and support for software development and maintenance were taken into account. Some question helped to determine if the prototype tool was helpful and effective to support software development and maintenance. Majority agreed on its overall usefulness.

After providing the importance and conformance of features, the score sheets must be analyzed and the best approach is determined. Based on the DESMET method, if the acceptance threshold is explicated, the analysis must be relied on the difference between the acceptance threshold fixed by the users and the score that each approach got for the feature. If the acceptance threshold cannot be achieved, the assessment should be based on the scores of the approaches relative to one another. As the acceptance threshold is not achievable in this research, the latter approach is used.

For simple feature, a score of “YES” or “NO” must be assigned. According to the DESMET suggestion, providing a simple feature need to score five, while score can be zero if it fails to provide a simple feature. Moreover, the importance assessment can be utilized as a weighting factor.

To express the analysis results, DESMET proposes numerically and graphically profile for this approach. Conclusively, the overall results for this approach are discussed. For numerical evaluation, the result based on DESMET method via the average evaluation profile for IM-DeCRuD is shown in the Table 5.

Subjects found that identification and handling processes of requirements components including crosscutting concerns using requirements boilerplates is a good element which is being supported by IM-DeCRuD tool. Despite of having some limitations in term of expressing users’ needs for a system to be developed or maintained, they acknowledged that by promoting common formats in system documentation, it will results in standardization and consistency besides claiming it can be seamlessly integrated in many development disciplines.

Composition feature which inter-relates crosscutting concerns and other requirements components was also regarded by the subjects. With systematic relationships between requirement components on top of having FUR as jointpoints to connect with other requirements components, it accommodates other features which come along with IM-DeCRuD tool.

Table 5. Average Evaluation Profile for IM-DeCRuD

Features	Importance	IM-DeCRuD
CI	HD	4
CS	HD	4
CTI	M	5
TS	HD	4
MP	HD	4
GUI	HD	YES
MT	HD	4
SC	HD	4

Another feature of identifying conflicts was highly acknowledged by the subjects. Good inter-relationship scheme between requirements components as well as practical implementation of Product-Oriented NFUR catalogue[16], requirements crosscutting prioritization scheme and Conflict Identification procedure accommodate conflicts identification at the earlier stage in software development and maintenance activities. Failing to deal with these conflicts might hinder software completion according to the schedule. With respect to traceability for requirements components, many of the subjects agreed that the tool able to help them to trace between requirements and design development phases by practically implementing the proposed mapping scheme. Using the same mapping scheme, crosscutting concerns and other requirements components identified at requirement phase were also agreed to be successfully mapped to class diagram at the design stage even in an upper level manner. This finding was seen to be a bit surprise to most of them as crosscutting concerns were previously regarded as a side note of system users’ requests and never being handled together with other requirements components.

Implementation of systematic requirements boilerplates for requirements components identification and handling as well as inter-relationship scheme that were addressed earlier were also being highly regarded by the subjects for simple and complex software maintenance support. Instead, the inter-relationship scheme as well as zooming capabilities (supported by the tool), make requirements components to have different levels of granularity that are deemed possible to support small and big scale software projects.

There were some good suggestions by the users. One of them suggested that this tool should be made open-sourced in order to facilitate vast sharing on its usefulness and improvements.

4. CONCLUSION

In this paper, verification and validation of the proposed approach and prototype tool are presented in term of qualitative findings by the domain experts. The results from myPolicy case study were evaluated on how the proposed approach can achieve its practicability in dealing with requirements identification, composition, conflicts identification, traceability, components mapping, software maintenance and project scalability. The scored results by the subjects were considered as the useful variables. Predicated on the results of the sessions, it is confirmed that the proposed approach provides some consequential achievements to handle the crosscutting concerns along with other requirements components. IM-DeCRuD's pragmatic composition scheme would be the platform for inspection process upon any conflicts of requirements that arises. Moreover, this scheme also assists in mapping all requirements components including crosscutting concerns from previous phases onto design stage where common industrial practises nowadays still do not equally regard crosscutting concern instead of other requirements components. Therefore, the subjects have the same opinion that the schemes used by IM-DeCRuD provide useful features that contribute to traceability process improvements as well as accommodate any scale of software maintenance projects.

REFERENCES

- [1] B. S. Ali and Z. M. Kasirun, "Developing Tool for Crosscutting Concern Identification Using NLP," Kuala Lumpur, Malaysia, 2008, p. IEEE.
- [2] B. S. Ali and Z. M. Kasirun, "A Review on Approaches for Identifying Crosscutting Concerns," Phuket, Thailand, 2008, pp. 855-859.
- [3] B. Tekinerdogan, C. Hofmann, M. Aksit, and J. Bakker, "Metamodel for Tracing Concerns Across the Life Cycle," Vancouver, Canada, 2007, pp. 175-194.
- [4] K. v. d. Berg, J. M. Conejero, and H. Juan, "Analysis of Crosscutting Across Software Development Phases Based On Traceability," in *Proceedings of the 2006 international workshop on Early aspects at ICSE Shanghai, China: ACM*, 2006.
- [5] F. Bübl, "Keeping Track of Crosscutting Requirements in UML Models via Context-Based Constraints," in *Companion to the 21st ACM SIGPLAN Symposium on Object-Oriented Programming Systems, Languages, and Applications* Portland, Oregon, USA: ACM, 2006.
- [6] E. Navarro and C. E. Cuesta, "Automating the Trace of Architectural Design Decisions and Rationales Using a MDD Approach," Paphos, Cyprus, 2008, pp. 114-130.
- [7] E. Navarro, P. Letelier, and I. Ramos, "Requirements and Scenarios: Running Aspect-Oriented Software Architectures," Mumbai, India, 2007.
- [8] J. Jasmis, S. Ibrahim, and R. B. Anom, "AORE/AOSD Approaches For Requirements and Design Crosscutting: A Review," in *International Conference on Software and Information Engineering (ICSIE 2011)*, Kuala Lumpur, Malaysia, 2011, pp. 591-596.
- [9] J. Jasmis, S. Ibrahim, R. B. Anom, and S. J. Elias, "IM-DeCRUD: An Approach For Requirements And Design Crosscutting Concerns To Support Software Evolution," in *Postgraduate Annual Research on Informatics Seminar (PariS 2012)*, Kuala Lumpur, Malaysia, 2012.
- [10] J. Jasmis, S. Ibrahim, S. J. Elias, R. A. Razak, and W. F. Abbas, "Criteria and approach implications for requirements and design crosscutting concerns to support software evolution," *Journal of Advanced Research in Computing and Applications*, vol. 6, pp. 1-8, 2017.
- [11] J. Jasmis, S. J. Elias, R. A. Razak, and W. F. Abbas, "Proposition of IM-DeCRuD Tool to Support Engineering Tasks for Requirements and Design Crosscutting Concerns," *Journal of Fundamental and Applied Sciences*, vol. 10, pp. 1080-1093, 2018/5/31 2018.
- [12] B. Kitchenham, S. Linkman, and D. Law, "DESMET: A Methodology for Evaluating Software Engineering Methods and Tools," *Computing & Control Engineering Journal*, vol. 8, pp. 120-126, 1997.
- [13] S. L. B. Kitchenham, D. Law, "DESMET: A Method for Evaluating Software Engineering Methods and Tools." vol. 21(1): SIGSOFT Notes, 1996, pp. 11-14.
- [14] K. Barbara Ann, "Evaluating Software Engineering Methods and Tool Part 1: The Evaluation Context and Evaluation Methods," *SIGSOFT Softw. Eng. Notes*, vol. 21, pp. 11-14, 1996.
- [15] K. E. Kendall and J. E. Kendall, *Systems Analysis and Design*, 4th ed.: Prentice Hall, 1998.
- [16] N. B. A. Chung L., Yu E., Mylopoulos J. , *Non-Functional Requirements in Software Engineering*. Boston, MA: Kluwer Academic Publishers., 2000.