CROSSCUTTING CONCERN IDENTIFICATION AT REQUIREMENTS LEVEL

Busyairah Syd Ali and Zarinah Mohd. Kasirun
Faculty of Computer Science and Information Technology
University of Malaya, 50603 Kuala Lumpur
bush@perdana.um.edu.my, zarinahmk@um.edu.my

ABSTRACT

An unresolved problem faced by software developers is the failure to identify and modularize certain artefacts that compose the software. It is difficult to modularize these artefacts because they are dispersed among other artefacts in the software properties. Aspects Oriented Requirements Engineering is showing encouraging results in improving identification, modularization and composition of crosscutting concerns. Identifying and documenting crosscutting concerns at the requirements-level is crucial. It avoids coupling between requirements, improves traceability among requirements, eases function modularization, reduces software complexity, enhances the correctness of the software design and most importantly it saves the cost. Although the research area is still in its infancy, several techniques for crosscutting concern identification have already been developed. However, all of the techniques reviewed are based on semi-automated way whereby human intervention is required to achieve the desired results. Therefore, in this paper, a fully automated technique based on Natural Language Processing (NLP) is proposed to identify crosscutting concern at the requirements level.

Keywords: Requirements Engineering, Aspect-oriented Requirements Engineering, Crosscutting Concern, Crosscutting Identification.

1.0 INTRODUCTION

Aspects Oriented Requirements Engineering (AORE) aims at addressing crosscutting concerns by providing means for identification, modularization, composition as well as analysis of their influence on other requirements in the specification documents. Mining crosscutting concerns involves large volume of specification documents. Documents such as interview transcripts are usually inaccurate, full of perceptible contradictions and missing vital information. Furthermore crosscutting concerns are often scattered across a document making their identification difficult. Making it worst, sometimes similar requirement occur in different parts of the document paraphrased in different words. Once identified, other factor such as interdependency or crosscutting influences between the concerns need to be analyzed to perform trade-off.

The goal of this research is to produce an automated approach to identify crosscutting concerns at the requirements level. In section 2, we present the definition of "crosscutting concerns" and "aspects". Section 3 provides the reason why we need to identify crosscutting concerns. Section 4 shows a review on related studies by several researchers. Section 5 provides the proposed solution on an automated approach to identify crosscutting concerns. Section 6 describes briefly the tool designed based on the proposed approach. Section 7 concludes the paper.

2.0 CROSSCUTTING CONCERN AND ASPECT

Sutton Jr and Rouvellou defined ***concern*** as "any matters of interest in a software system" [11]. It can be directly related to the system or its environment. We can also define a requirement as a concern stated by the system users or stakeholders. They added crosscutting concern can either be functional or non-functional requirements.

When a concern crosscuts one or more of other concerns, they are called ***crosscutting concern***. For example, in case of two requirements 'A' and 'B', an act of software enhancement is initiated in which 'B' cannot be satisfied without affecting 'A' that means requirement 'A' crosscuts requirement 'B'. In this case the requirement that crosscuts others are referred to as being crosscutting concern, which is 'A'. A number of researchers also address crosscutting concern as ***candidate aspect*** [7]. Crosscutting concern is tangled and scattered in nature. This is illustrated in Fig. 1. Some crosscutting concerns can be obvious and can be easily identified. But often crosscutting concerns are subtle. So it is difficult to identify them. Once they are identified, they are encapsulated into modules called ***aspect***. ***Crosscutting influence*** indicates the relationship between two or more requirements which is established by one crosscutting the other [8]. For example if 'A' has crosscutting influence on 'B' it means 'A' crosscuts 'B'. Crosscutting influence denotes dependency among requirements. However not all dependencies are of crosscutting nature.

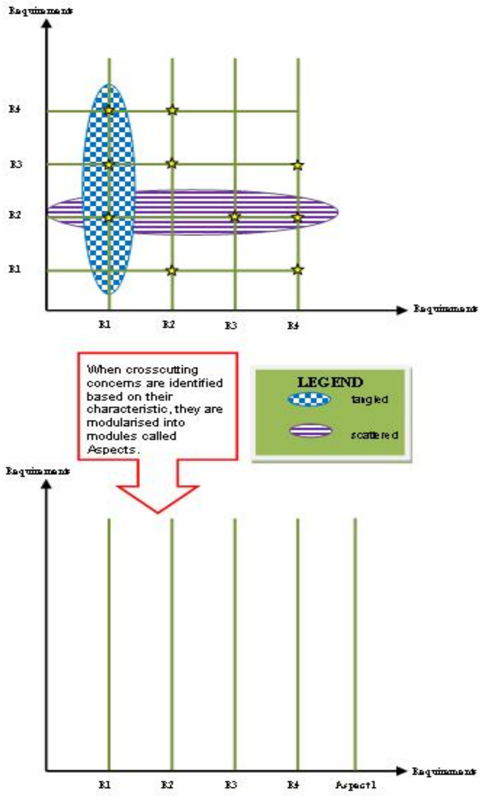

Fig. 1: An Illustration of Crosscutting Concern

3.0 EARLY ASPECTS

During the requirements analysis phase, it is difficult to see how the requirements are influenced by each other and how it will impact the whole software development process. If such concerns are not identified and modularized soon enough it will affect the choice of software architecture. Then, it will be too late to reverse to the earlier process. This will cause more cost, time and effort. Therefore, it is important to identify crosscutting concern at the early stage so that it is not overlooked in the subsequent phases. Having the aspects nicely encapsulated, it is easy to trace the effects on other requirement when there are any changes made to the software system. This is because by nature aspects do influence other requirements. Furthermore, aspects can either be a non-functional or functional requirement that will ensure the completeness of the software system. Neglecting these concerns leads us to developing incomplete software with poor correctness.

Comprehensive documentation of crosscutting requirements and crosscutting influences facilitates easier requirements evolution. For example if a requirement crosscuts other requirement and is ambiguous, then the requirement needs to be changed to have a clear cut its role. Maybe at later stage, if not identified, it can appear as a bug in the software system. So, when do we identify them? We would say the earlier the better.

4.0 RELATED RESEARCHES

Although the necessity of identifying requirements-level crosscutting concerns has been stated, only few attempts have been made so far as how to systematically identify them. Descriptions on each approach identified emphasizing on the flow and steps to mine crosscutting concerns, advantages and disadvantages of each approach are given below.

4.1 Theme / Doc Approach

Theme/Doc Approach [5] provides a semi-automated identification of crosscutting concerns in requirements specification documents. The approach is based on lexical analysis. Firstly the requirements in the specification document must be numbered. Then the developer needs to go through the whole document to identify a set of action words presented as 'theme' in this approach. The themes will be classified into 'major action' and 'minor action'. The minor actions can be subtheme for the major actions. Next the relations between the themes and the requirements are mapped using 'Action View Model'. This has to be done manually by the developers. The themes T1...Tn are illustrated using diamond shape and the requirements R1...Rn in rounded box shape. If a theme is mentioned in a requirement, then a line presents the link between them. Fig. 2 below illustrates the Action View Model that maps the relation between a set of themes identified and the requirements.

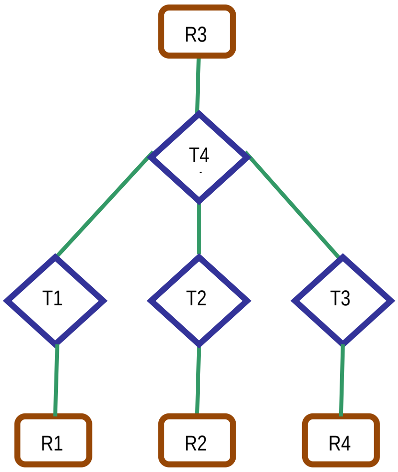

Fig. 2 : Action View Model

The set of actions identified in the above scenario are T1, T2, T3 and T4. The model shows that requirement R3 is shared by all the themes having T4 overlaid on T1, T2 and T3. It means T1, T2 and T3 are behaviorally relying on T4. Therefore T4 is identified as a crosscutting concern. It also shows that T1, T2 and T3 are directly associated with R1, R2 and R4 accordingly.

The focus of this approach is on the requirements shared by more than one theme. If a requirement is shared by two or more themes, the decision on which theme should provide the functionality should be made. If two or more themes are relying and overlaid on each other, than the crosscutting concern is identified. This method allows all the relationship between the requirements to be clearly identified and mapped while ambiguity in the requirements can also be seen. However, this approach is only applicable for structured requirements document. As for the developers, they must possess the domain knowledge. Hence they must go through the whole requirements source document to identify the crosscutting concerns. They have to manually map the relationship between the themes and requirements. It is costly and time consuming to handle large amount of requirement sources. The developer must ensure the cardinality between the themes and requirements is at the most simplest form to identify the crosscutting concern.

4.2 Early Aspect Identification Method

Sampaio [3] and Rashid [1] utilizes Corpus-based Natural Language Processing (NLP) to enable identification of crosscutting concerns in a semi-automated way. This approach enables the requirements engineer to automatically mine the requirements from structured or unstructured sources to identify and build a structured aspect-oriented model of the requirements. The mining tool developed based on the approach is EA-MINER [2&10]. WMATRIX [9] a NLP processor is used as part of the EA-Miner tool for the specific purpose of identifying the crosscutting concerns and the relationship between the requirements. WMATRIX provides features such as part-of–speech and semantic tagging, frequency analysis and concordance to identify domain concepts and potential significance. The part-of-speech module automates the extraction of nouns and verbs from the text while the semantic tagging module groups related words and multi-word expressions into related domain. The process begins by having the developer manually feeding any type of requirements document to the EA-miner Tool. The EA-Miner tool then reads the files and passes them to WMATRIX. Then WMATRIX identifies concerns and viewpoints using Natural Language Processing (NLP) technique. EA-Miner enables tailoring of information that flows in and out the WMATRIX. Then an Intermediate Model (Viewpoint) is produced. Next the EA-MINER will set criteria to filter the results from the intermediate model to identify the candidate aspects and generate an Aspectual Model showing the relationship between the requirements. WMATRIX examines the dispersion of candidate aspects across the various requirements. If it appears well dispersed in many requirements then the case for that candidate aspect is stronger.

4.3 Information Retrieval Based Technique

Information Retrieval Technique [8] for crosscutting identification is based on subjective assumption by the system analyst. This approach is only applicable to the functional requirements crosscut by some response-time requirements. Therefore it cannot be used as a general approach to identify all crosscutting concerns or crosscutting influences.

4.4 Identification of Crosscutting Concerns with UML

Brito [6] and Araujo [7] introduced an approach to handle crosscutting non-functional concern at the requirements level using Unified Modelling Language (UML) model. Fig. 3 below shows the model for this approach. This approach has three main steps. Firstly all the functional and non-functional requirements are identified. Then the crosscuts among the non-functional requirements are identified. Next the functional requirements together with the aspects identified are composed and modelled into UML. Then the conflicts are identified and resolved.

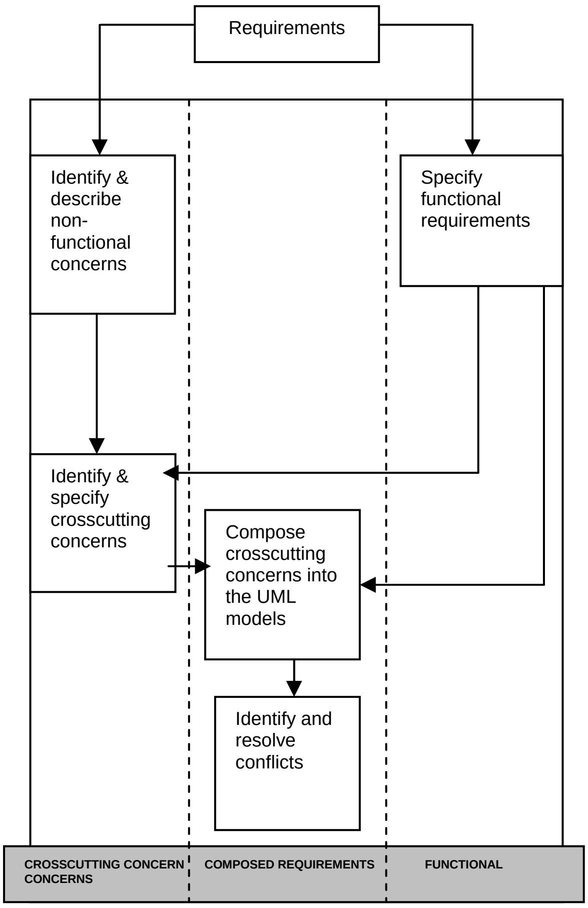

Fig. 3: Model for Aspects Oriented Requirements with UML

This approach adopts the concepts of *overlapping, overriding and wrapping* to define the composition part of the model. The concepts are described as below:

- **Overlapping:** the requirements of the aspect modifies the functional requirements they transverse. In this case, the aspect requirements may be required before the functional ones, or they may be required after them.
- **Overriding**: the requirements of the aspect superpose the functional requirements they transverse. In this case, the behavior described by the aspect requirements substitutes the functional requirements behavior.
- **Wrapping:** the requirements of the aspect "encapsulate" the functional requirements they transverse. In this case, the behavior described by the functional requirements is wrapped by the behavior described by the aspect requirements.

4.5 Summary of Related Works

All the methods described above works in a semi-automated way to identify crosscutting concerns at the requirements level. However each method has its own plus and minus features. We identified Theme/Doc method as the best method to identify relationships between the concerns. However this approach is not suitable to handle complex problems and large amount of requirements sources. The Early Aspects Identification method using Corpus –based Natural Language Processing (NLP) is a good approach because it is applicable for any type of requirements document despite the structure. Unlike other method, they only accept structured documents. Information Retrieval Based Technique is a promising method for voluminous document because the scope for the type of crosscutting concern is only one. But it is not applicable to identify all crosscutting concerns in the requirements document. The last method, Identification of Crosscutting Concern with UML is a simple method and applicable to small scale requirements. Unfortunately it can only identify non-functional crosscutting concerns. In a nutshell all the methods explained above require human intervention to produce the final model. Based on the studies on all the existing methods for crosscutting concern identification, we found that each method has its own advantages and disadvantages. Our earlier research work [4] has adopted and integrated certain techniques from the existing approaches to develop a fully automatic crosscutting concern identification model. Due to the complexity to analyze the output of the proposed NLP tool, WMATRIX, the model is revised in this paper to use a general post tagger to meet the purpose.

5.0 REVISED MODEL FOR AUTOMATED CROSSCUTTING CONCERN IDENTIFICATION USING NLP

The revised model for Automated Crosscutting Concern Identification using NLP or CCCINLP is depicted in Fig. 4. The CCCINLP still maintains all the components that compose the model which is described in detail in [4]. However, the implementation of task 3 and 4 which is presented in section 5.3 and 5.4, respectively, will be supported by general tagger.

5.1 Structure Requirements

This task involves numbering all the requirements agreed by the stakeholders. This is required to identify and manipulate each requirement uniquely in the next stages.

5.2 Remove Redundancy

Sometimes different stakeholders tend to specify the same requirement more than once. In order to eliminate duplication of requirements, the redundant requirements are removed in this task.

5.3 POS Analysis

This task is to extract verbs from each requirement. The frequency of their occurrence will be calculated to show its dispersion throughout the document. Higher level of dispersion indicates the strength of the verb as candidate aspect. Corresponding verbs will be used for modelling the relation with the requirements and interdependency among other verbs.

5.4 Semantic Analysis

This task utilizes semantic tagger to analyze the context of the phrase in which the verb is used. This information is used to identify verbs used to describe similar requirements.

5.5 Filter Verbs Identified

Based on the semantic analysis performed, duplication of the verbs in terms of the context is discarded. For example the usages of 'protect' and 'secure' in the same context allow us to discard one of the term since it refers to the same meaning.

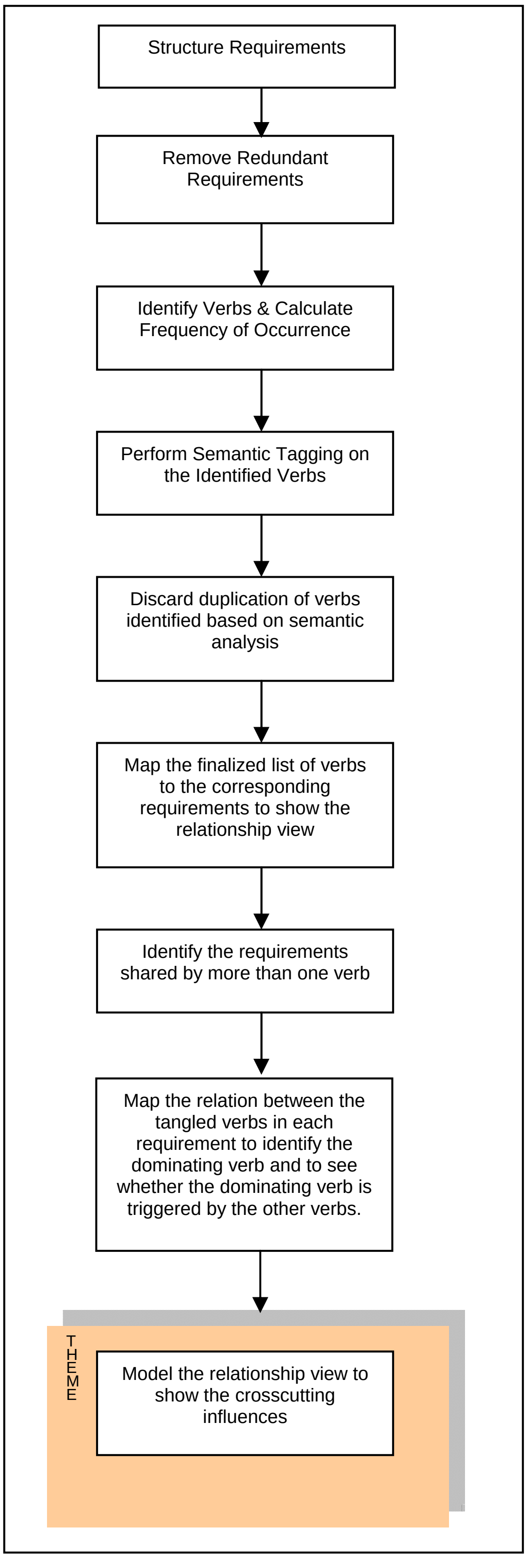

Fig. 4: CCCINLP Model

5.6 Map Relationship View

To narrow down the scope of identification of crosscutting concern, we map the requirements using a matrix as shown in Table 1 to identify the requirements {R1..Rn} influenced by corresponding verbs {v1..vn}. For example R4 is influenced by v1, v2, v3 and v4. Therefore there are descriptions of four verbs tangled within requirement R4. The matrix can become quite large if there are many requirements and many verbs. This can be mitigated by imposing constraints in the next stage.

Table 1: Matrix Mapping the Relationship View

Verbs / Requirements	v1	v2	v3	v4	…	vn
R1				✔		
R2	✔					
R3		✔				
R4	✔	✔	✔	✔		
…						
Rn						

5.7 Refining the Relationship View

Based on the relationship view, the requirements shared by more than one verb and the scattered verbs are identified. As shown in Table 1, R4 is an example. The matrix is then refined by showing the requirement shared by more than one verb and all the requirements influenced by the verbs identified in the shared requirement. The refined matrix is tabulated in Table 2.

Table 2: Matrix Mapping Refined Relationship View

Verbs / Requirements	v1	v2	v3	v4
R4	✔	✔	✔	✔

Table 3: Identifying Dominating Verb

Verb	v1	v2	v3	v4
v1		✔	✔	✔
v2				
v3				
v4				

5.8 Dominant Verb

A matrix as shown in Table 3 is regenerated to map the relationship between the tangled verbs in each requirement to identify the dominating verb in the requirement and to see if the dominating verb is triggered by the other verbs. The dominating verb is the candidate aspect.

5.9 Modelling Crosscutting Influences

The requirement and verbs identified as the candidate will be modelled to identify the crosscutting concern using Action View Model as used in Theme/Doc approach. The verbs v1...vn are illustrated using diamond shape and the requirements R1...Rn in rounded box shape. If a verb is related to a requirement, then a solid line presents the link between them. Fig. 5 illustrates the initial relationship view that maps the relationship view between the verbs and the corresponding requirements. The relationship view is non-hierarchical, so even though it looks as though some verbs are "higher" than others, this is just a coincidence of layout.

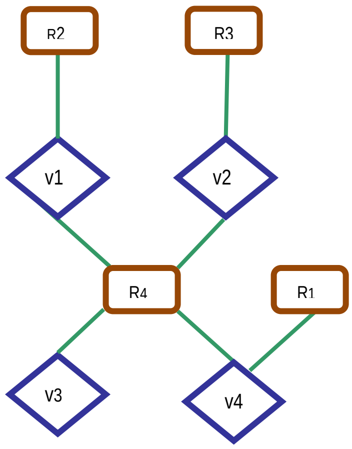

Fig. 5: Initial Relationship View

Fig. 5 shows the relationship view for the above scenario. Each requirement (R1..R4) is linked to the verb mentioned in the requirement. R3, for instance, mentions v2 and so is linked to it in the view. Some requirements refer to more than one verb. R4, for instance refers to four verbs: v1, v2, v3 and v4. The relationship views can become quite large if there are many requirements and many verbs. From the view, it is known that the candidate aspect resides in R4. Hence by identifying dominating verb, the final view, Crosscutting Relationship View is regenerated as shown in Fig. 6.

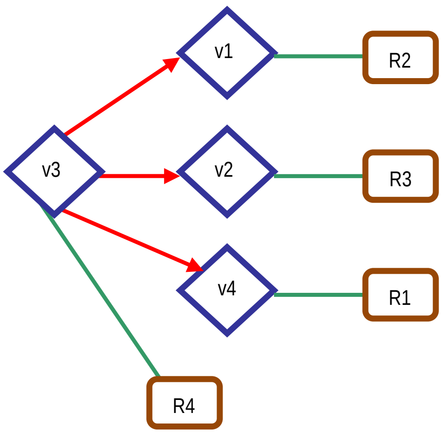

Fig. 6: Crosscutting Relationship View

6.0 C3I TOOL

The approach above is realized by designing a tool, called C3I based on the CCCINLP model. 3CI Tool is a web-based tool that processes text-based requirements documents written in English language to identify crosscutting concerns. It is developed to verify the CCCINLP method developed in this research. This tool is meant to aid the system developers to effectively analyze the requirements document for system design and implementation. The tool inputs text-based requirements documents and processes the requirements using Part of Speech Tagging (POST) module incorporated within the tool. The preliminary design of the tool has been described by Ali and Kasirun [4]. The following are the functional requirements of the tool:

- It shall accept any English-text-based requirements document as an input.
- It shall identify each requirement uniquely.
- It shall be able to extract requirements clauses having more than one verb phrases.
- It shall be able to calculate the frequency of occurrence of identified verbs in the requirements document.
- It shall be able to identify the concern with scattering behaviour if there should exist any concern as such in the requirements document based on the frequency analysis.
- It shall perform Dominant Verb Analysis on each requirement clause having more than one verb.
- It shall incorporate POST module as part of the system.
- It shall have the capability to perform part of speech analysis to tag each word in the requirements clauses using the POST module.

The successfulness of C3I tool in offering all these features would contribute on the effectiveness of the CCCINLP model.

7.0 FUTURE WORKS

The paper has presented current researches conducted on crosscutting concern identification at requirements level and analyzed the advantages and drawbacks of each approach. Based on the analysis made a new model CCCINLP is described in this paper. The future work will concentrate on the development and evaluation of the C3I tool developed based on this model.

REFERENCES

[1] A. Rashid, A. Moreira, and J. Araujo. “Modularisation and Composition of Aspectual Requirement”s. In *Proceedings of the 2nd International Conference on Aspect Oriented Software Development*, pages 11–20. ACM Press, 2003.

[2] A. Sampaio, R. Chitchyan, A. Rashid, P. Rayson. “EA-Miner: A Tool for Automating Aspect-Oriented Requirements Identification”. *International Conference on Automated Software Engineering (ASE'05),* Long Beach, California, USA, November 7-11, 2005.

[3] A.Sampaio, A. Rashid, P. Rayson. “Early-AIM: An Approach for Identifying Aspects in Requirements”. Poster paper at *Requirements Engineering Conference,* Paris, France, 2005

[4] B. S. Ali and Z. M. Kasirun “Developing Crosscutting Concern Identification Tool using NLP”. In *Proceedings of International Symposium on Information Technology*, Malaysia, 2008.

[5] E. Baniassad and S. Clarke. “Finding Aspects in Requirements with Theme/Doc”. In *Proceedings of Early Aspects 200*4: (AOSD 2004) Lancaster, UK, 22 Mar, 2004.

[6] I. Brito. “Aspects Oriented Requirements Engineering”. In *Proceeding of the7th International Conference on Unified Modelling Language (UML)* Doctoral Symposium, 2004.

[7] J. Araujo, A. Moreira, I. Brito and A. Rashid. “Aspect-Oriented Requirements with UML”. Workshop *on Aspect-Oriented Modelling with UML* (held with UML 2002).

[8] L. Rosenhainer. “Identifying Crosscutting Concerns in Requirements Specifications”. In *Proceedings of OOPSLA Early Aspects 2004: Aspect-Oriented Requirements Engineering and Architecture Design Workshop*. October 2004, Vancouver, Canada.

[9] P. Sawyer, P. Rayson, and R. Garside. "REVERE: Support for Requirements Synthesis from Documents." *Information Systems Frontiers Journal*. Volume 4, issue 3, Kluwer, Netherlands, pp. 343 – 353, 2000.

[10] R. Chitchyan, A. Sampaio, and A. Rashid "A Tool Suit for Aspect-Oriented Requirements Engineering" *Workshop on Early Aspects* (held at ICSE 2006), Shanghai, China.

[11] S. M. Sutton Jr and I. Rouvellou. "Modeling of Software Concerns in Cosmos". In *Proceedings of the 1st International Conference on Aspect-Oriented Software Developme*nt, pages 127–133. ACM Press, 2002.

ACKNOWLEDGMENTS

Sincere thanks to the University of Malaya for sponsoring this research via the fundamental research fund (Account No.: FP067/2007A).

BIOGRAPHY

Busyairah Syd Ali received B.IT degree from Multimedia University, Malaysia in 2003. She is currently completing her master's degree in Software Engineering in University of Malaya, Malaysia. She has served as an engineer in Telekom Malaysia for four years. Her research interest includes surveillance systems, navigation systems, air traffic control systems, database and requirements engineering. She has published three conference papers.

Zarinah Mohd. Kasirun is a senior lecturer in the Department of Software Engineering at the Faculty of Computer Science and Information Technology, University of Malaya. She teaches software engineering modules at both the undergraduate and master levels. Her research areas include Requirements Engineering, Computer-Supported Collaborative Learning, Object-Oriented Analysis and Design.