Process to Identify the Crosscutting Concerns in Changing Requirements through Aspect-Oriented Software Engineering

Hema Kashyap
Sharda University
Greater Noida

ABSTRACT

The traditional software development methodology now cannot meet the expanding software scales, changing software requirement and the software evolution processes very well. Aspect-Oriented Software Engineering (AOSE) helps to meet out these entire problems during software development. With the advancement in the field of software engineering, software requirements are changing at a fast pace which leads to the modification of the existing systems but the problem arises after more and more modifications. Due to this software system becomes more complex to understand and maintain. AOSE approach provides new mechanisms to identify the crosscut elements of a changing requirements specification. In this paper we present a process to identify crosscutting concerns in changing requirements through the AOSE and also represent these crosscutting concerns along with the already existing requirements by making use of Unified Modeling Language (UML) diagram.

Keywords

Aspect Mining, Reverse Engineering, Reengineering.

1. INTRODUCTION

Software engineering starts with the requirements to the implementation. During requirements number of problems is faced by the software developer due to changing requirements, expanding the software scales and software evolution during the software maintenance. Hence, dealing with legacy software in some or other context becomes complex due to less understandability. AOSE provides the systematic means for identification, modularization, representation and composition of crosscutting concerns. With the emergence of the AOSE, these problems are solved to some extent. AOSE deals with identification of crosscutting concerns and encapsulating them into aspect. Crosscutting concerns refers to the concerns which are scattered and tangled.

The crosscutting concerns identified at the requirement and the architectural level are termed as early aspects. Such concerns include security, availability, mobility and real-time constraints. Early detection of crosscutting concerns at the early development stages of requirements engineering and architectural design enables effective modularization of concerns. The lack of modularization of such concerns/properties can result in a large ripple effect on other requirements and architectural components upon evolution. The body that work on these separations of concerns in the requirements engineering includes viewpoints and use cases. Use case diagrams provide a systematic means for handling such concerns.

General model for Aspect-Oriented Requirement Engineering (AORE) was proposed by [1]. The model separates the functional and non-functional requirements. Non-functional requirements are global properties of a system that constrain the functional requirements [2]. Further identifies the crosscutting concerns from each type of requirements which are then modeled using UML. In this way issues are resolved which occur as a result of crosscutting concerns.

The model proposed is for existing requirements which come from the customer but with the changing needs, technology requirements also changes and for these changing requirements developing the module separately may further leads to the crosscutting concerns hence there's a need of such model which deals with the changing requirement as soon as they come crosscutting concerns must be identified and implemented in UML model so that in the later phases of software development they may not lead to scattering of concerns.

2. RELATED WORK

In last couple of years this area has been growing interest in propagating the aspect paradigm. Integrating aspects with UML seems to be an obvious area of research. Suzuki and Yamamoto proposed an extension to UML to support aspects, where an aspect is described as a classifier in the meta-model [3]. They also proposed a XML based aspect description language to interchange aspect models between development tools such as CASE tolls and aspect weaver [3]. They also proposed a XML based aspect description language to interchange aspect models between development tools such as CASE tolls and aspect weaver. The aspect-oriented requirements engineering approach by Grundy is targeted to component based software development, where there is a characterization of diverse aspects of a system that each component provides to end users or other components [4].

An UML compliant approach to handle quality attributes at the requirements activity of the software development process was proposed in [5]. A composition pattern is an approach to handle crosscutting requirements at the design level [5].

This approach promotes reusability and traceability to the following activities of the software development. This model is based on subject-oriented design and uses UML templates.

The work presented in this paper is the extension of the requirement model [6] for the changing or new requirements that arises with time.

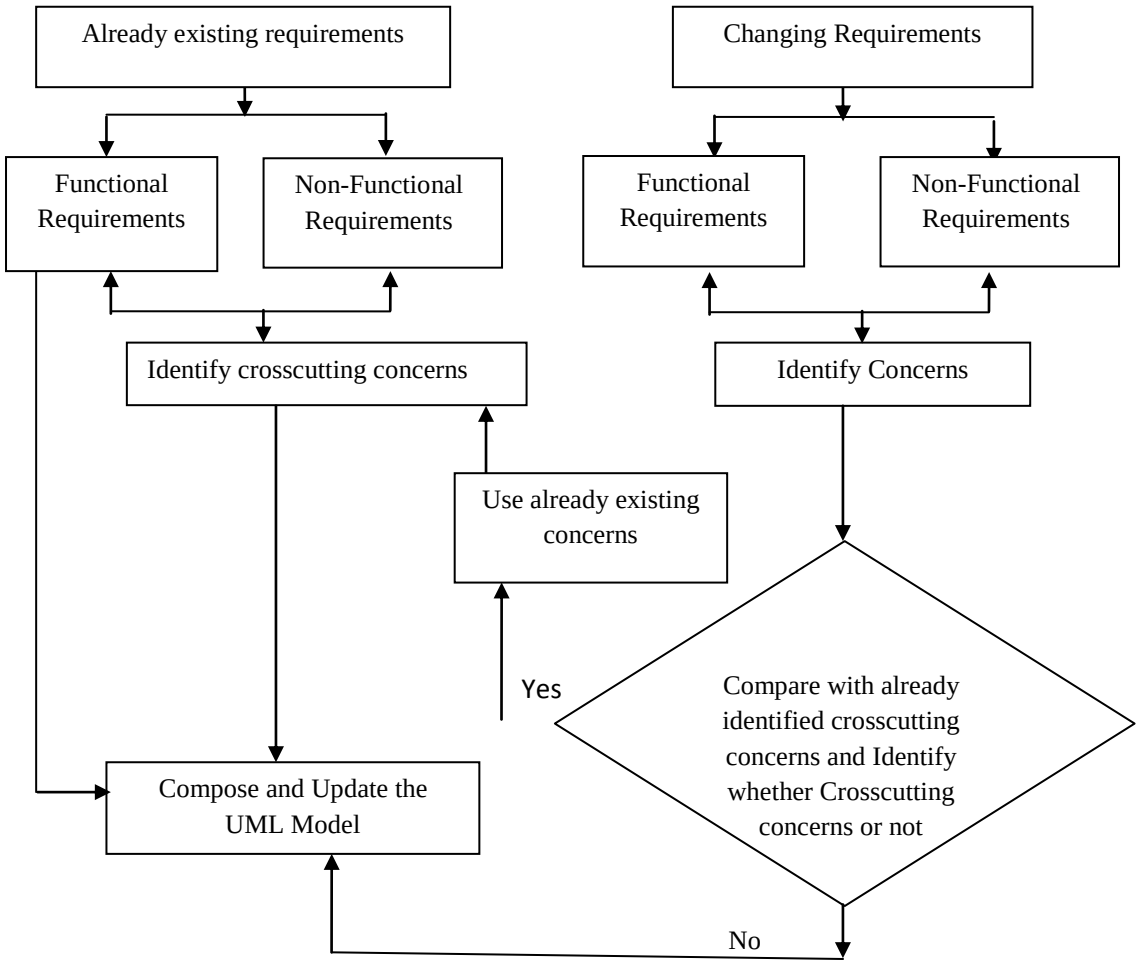

Fig 1: Aspect -Oriented Requirement Model for Changing Requirements

3. ASPECT-ORIENTED APPROROACH FOR CHANGING REQUIREMENTS

The proposed approach is to deal with the changing requirement which arises as a result of technology change, software enhancement and evolution. This paper describes a process which presents the changing crosscutting requirements with the UML.

This approach can be further partitioned into four parts:

- **Crosscutting concerns:** This includes identification of the crosscutting concerns from the non-functional requirement. Functional concerns: This includes the functional specifications with the help of UML model.
- **Composed requirements:** this includes composition of both the functional requirements modeled using UML with the aspects.
- **Resolving Issues:** this includes the conflicts that may arise due to composition process. The issues are dealt based on three concepts overlapping, overriding and wrapping.

The approach presented in Figure 1 could be discussed under two heads given as follows:

3.1. For Existing Requirements:

- This section deals with the existing requirements that come from the customer at the time of development of the software. The process includes identification of the crosscutting concerns from the functional and non-functional requirements. These crosscutting concerns are further represented with the help of UML model. This part of approach includes following steps:
- The existing requirements are those requirements that come from the customer at the time of software development. These requirements include the features that must be present in the software developed.
- These requirements can be further categorized into functional and the non-functional requirements. Functional requirement defines what a system is supposed to do whereas non-functional requirement defines how a system is supposed to be.
- Followed by categorization both type of requirements are scanned for the crosscutting concerns.
- If the crosscutting concerns exist, compose the UML model.
- For rest of the concerns that is non-crosscutting concerns are included in the UML model and are updated accordingly.

3.2. For New Requirements:

- This section deals with the changing requirements that arises with the time and is to be included in the software. The concerns are identified and are then compared with

the already identified crosscutting concerns. It includes following steps:

- With the advancement in technology, changing needs or system evolution new requirements evolve. These are the new requirements which are to be included in the already existing software.
- These new requirements are further categorized into functional and non-functional requirements as done with the previous requirements.
- The various concerns are identified from these changing requirements and are specified separately.
- Then these identified concerns are compared with the already identified crosscutting concerns of the existing requirements.
- If the same concerns exist then the already identified crosscutting concerns are used rather using them separately which increases the complexity and leads to scattering of concerns.
- If the identified concern doesn't match with the crosscutting concerns then for these concerns UML model is updated.

4. CONCLUSION

This paper modifies an approach to handle the crosscutting concerns at the requirement level, using UML for the changing requirements. This approach classifies the requirement to functional and non-functional requirements which further identifies the various concerns from the new requirement. The concerns are further compared with the crosscutting concerns of the existing requirement and the UML model is updated likewise

5. REFERENCES

[1] Rashid, A., Sawyer, P., Moreira, A., Araujo, J., (2002). "Early Aspects: A Model for Aspect-Oriented Requirement Engineering", proceedings of the IEEE Joint International Conference on *Requirement Engineering", pp.199-202.*

[2] Chung, L., Nixon, B., Yu, E., Mylopoulos, J., (2000). "Non-Functional Requirements in Software Engineering", International Series in *Software Engineering*, Vol. 5, *pp.476.*

[3] Suzuki, J. and Yamamoto, Y., (1999). "Extending UML with Aspects: Aspect Support in the Design Phase", submitted to the 3rd *Aspect-Oriented Programming* Workshop, *pp.1-6.*

[4] Grundy, J., (1999). "Aspect-oriented Requirements Engineering for Component-based Software Systems", proceedings of the 4th IEEE International Symposium on *Requirements Engineering, pp. 84-91*.

[5] Moreira, A., Araújo, J., Brito, I., (2002), "Crosscutting Quality Attributes for Requirements Engineering", proceedings of the 14th International Conference on *Software Engineering and Knowledge Engineering, pp. 167-174.*

[6] Araujo, A.Moreira, Brito, I., and Rshid, A., (2002). "Aspect-oriented Requirement with UML" proceedings of the International Workshop on Aspect-Oriented Modelling

[7] Kashyap, Hema. "Aspect based reverse engineering process." In Engineering (NUiCONE), 2012 Nirma University International Conference on, pp. 1-3. IEEE, 2012.