GPRN: A Hierarchical Framework for Aspect-oriented Requirement Modeling

Chengwan He, Chengmao Tu
School of Computer Science and Engineering, Wuhan Institute of Technology, Wuhan 430073, China
Hubei Province Key Laboratory of Intelligent Robot, Wuhan 430073, China
hechengwan@hotmail.com

Abstract

One of the most important characteristics in aspect-oriented requirement modeling is effectively dealing with crosscutting concerns. This paper presents a hierarchical GPRN framework for aspect-oriented requirement modeling. The framework breaks requirements into three layers including goal layer, process layer and requirement net layer. Goal layer defines crosscutting concerns from goals, process layer defines when crosscutting concerns affect core concerns, and requirement net layer defines how crosscutting concerns affect core concerns. Meta-models are proposed to guide the modeling of different layers. This framework provides a systematic method that defines requirements of aspect-oriented software.

Keywords: Aspect-oriented Software, Requirement Modeling, Meta-model

1. Introduction

Aspect-Oriented Programming (AOP) [1] approach represents software system in terms of different independent concerns such as functional requirements, non-functional requirements and platform characteristics, and well modulates crosscutting concerns. AOP is regarded as necessary complement of Object-Oriented Programming.

As shown in Figure 1, the AOP language develops rapidly, and many approaches and techniques such as reuse of aspect and definition of crosscutting concern are proposed by most of researchers at implementation phase. However, where are the crosscutting concerns at implementation phase from? Whether could the crosscutting concerns at implementation phase be identified at earlier development stages such as requirement and design phase? How are the crosscutting concerns of requirement phase converted into the aspects of design and implementation phase? Obviously, aspects needed to be managed at a higher phase. These issues are noticed by some researchers and the concept of early aspects [2] is proposed to represent aspect-oriented requirement engineering and architecture design. Many research results are reported by international authoritative journals such as IEEE Software [2] and the top international conferences such as ICSE (the International Conference on Software Engineering) [3][4], international conference on AOSD (Aspect-Oriented Software Development) [5],

Figure1. Aspect-oriented software development stages

As shown in Figure 1, the AOP language develops rapidly, and many approaches and techniques such as reuse of aspect and definition of crosscutting concern are proposed by most of researchers at implementation phase. However, where are the crosscutting concerns at implementation phase from? Whether could the crosscutting concerns at implementation phase be identified at earlier development stages such as requirement and design phase? How are the crosscutting concerns of requirement phase converted into the aspects of design and implementation phase? Obviously, aspects needed to be managed at a higher phase. These issues are noticed by some researchers and the concept of early aspects [2] is proposed to represent aspect-oriented requirement engineering and architecture design. Many research results are reported by international authoritative journals such as IEEE Software [2] and the top international conferences such as ICSE (the International Conference on Software Engineering) [3][4], international conference on AOSD (Aspect-Oriented Software Development) [5],
international conference on requirements engineering [6][7]. However, these are just at early stage of research and especially research on AORE is not adequate.

Effective method of requirement modeling contributes to defining more complete and consistent software requirement. Traditional methods of requirement modeling such as viewpoints [8], use cases [9], goals [10] have deficiency in explicitly addressing crosscutting concerns and are difficult to effectively adapt to aspect-oriented software.

One of the most important characteristics of aspect-oriented requirement modeling is to effectively address crosscutting concerns and solves the three questions about what (which concerns are the crosscutting concerns), when (when crosscutting concerns affect core concerns) and how (how crosscutting concerns affect core concerns). In recent years, though some researchers have achieved some preliminary results [2][11][12][16], there are still some shortcomings among them.

This paper presents a hierarchical GPRN framework for aspect-oriented requirement modeling. The framework breaks requirements into three layers including goal layer, process layer and requirement net layer. Goal layer defines crosscutting concerns from goals, process layer defines when crosscutting concerns affect core concerns, and requirement net layer defines how crosscutting concerns affect core concerns. Requirement net could be further refined and verified at the design stage. For example, it can be verified that whether precondition and postcondition of process are satisfied after weaving aspect.

2. Related work

Baniassad et al [2][11][12] proposes an approach of identifying and defining aspects at requirement phase. Non-functional properties are identified as aspects and a set of informal combination rules are provided. The approach comprises three steps including identifying action and entity from requirement specification, classifying action by themes and identifying crosscutting themes. The approach focuses on the needs of identification of crosscutting concerns at requirement phase but lacks of systematic method of building requirement model.

Reference [13] proposes an approach using natural language processing to identify crosscutting concerns based on verb phrase in the specification. Rashid et al [14] provides an AORE model used to identify, define and assess aspects at requirement phase. The approach comprises seven steps including identifying and defining concerns and stakeholders’ requirements, defining relations between requirements, representing aspects, defining candidate aspects and combination rules of requirements, identifying and addressing conflicts between candidate aspects, redefining requirements and defining the dimension of aspects.

Reference [15] proposes an approach supporting the modeling of crosscutting by extending KAOS. I. Brito et al [16] provides an approach integrating the non-functional framework into the requirement model of aspect-oriented software. The approach consists of four tasks including identification of concerns, definition of concerns, definition of crosscutting concern and definition of combination of concerns.

Reference [17] proposes an approach based on object constraint language (OCL) which supports domain model of aspect-oriented software. The approach combines the model represented in xml with constraint represented in OCL to form the model with constraint.

Reference [19] presented a framework RGPS (Role, Goal, Process, Service) for requirement modeling of the networked software. Role meta-model is used to guide abstraction of the hierarchical structure of the organizations in the domain, goal meta-model is responsible for instructing extraction and refinement of goals, process meta-model is responsible for achieving modeling of the business process to realize goals, service meta-model is responsible for instructing selection and composition of the services.

Zenon Chaczko et al [20] proposes a new approach for developing software requirements analysis CASE tools specifically intended to suit the particular needs of cross-time-zone development projects. Yang et al [21] proposes a meta-model to structuralize requirements, and argues that stakeholders can use ontologies to fill in the meta-model to composite a requirement summary. Guo [22] proposes an approach to analyze the software requirement stability based on the regression analysis and control chart analysis, and a web-based requirement stability analysis tool has been implemented.
3. GPRN framework for requirement modeling

Characteristics of aspect-oriented requirement are various goals, complex business logic and widespread influence caused by crosscutting concerns. On the basis of RGPS [19], we propose an aspect-oriented requirement modeling framework GPRN, as shown in Figure 2. Requirement model consists of three layers including goal, process and requirement net in GPRN. Requirements are decomposed from top to bottom with guidance of meta-models represented in each layer.

Each meta-model includes the representation of crosscutting concerns as well as the relation between crosscutting concerns and core concerns. Crosscutting concerns in each layer are defined as crosscutting goal, crosscutting process, aspect net respectively. Crosscutting goal defines which concerns are the crosscutting concerns (what), crosscutting process defines when the crosscutting concerns tangle with the core concerns (when), and aspect net defines how the crosscutting concerns affect the core concerns (how).

3.1. Goal meta-model

Goal meta-model is shown in Figure 3.

The user’s goal is usually abstract and in higher level, it should be refined or decomposed to guide subsequent development. Refinement of goal ends with a collection of operational goals. Operational goal corresponds directly with one process and is a functional goal. Crosscutting goal affects the functional goal. The goal decomposition is similar to feature decomposition. Four kinds of relation between high-level goal and low-level goal are mandatory, optional, alternative and OR. Three kinds of relation between crosscutting goal and functional goal are before (crosscutting concerns should be
executed before the functional goal), *after* (crosscutting concerns should be executed after the functional goal) and *around* (crosscutting concerns may surround the functional goal). Goal layer aims mainly to guide to differentiate types of goals such as functional goal, non-functional goal, crosscutting goal, operational goal etc, refine goal and determine the way of refinement and dependencies between goals etc.

### 3.2. Process meta-model

Process meta-model is shown in Figure 4.

![Figure 4. Process meta-model](image)

Process consists of atomic process, composite process and crosscutting process. Composite process consists of at least one control structure but atomic process not. Crosscutting process affects atomic process. Process consists of input, output, precondition and postcondition. Input and output represent the data flow converted in process. Precondition and postcondition represent respectively the beginning and end state held in process. Process layer aims mainly to guide to combine process based control structure and specify the input, output, precondition and postcondition of atomic process, scope of crosscutting process and relation with goal meta-model.

### 3.3. Requirement net meta-model

Petri Net [18] is a formal method for modeling and verifying software. Definition of Petri Net in [18] is following.

**Definition of Petri Net**: A Petri Net is a 3-tuple \( PN = (P, T, F) \) where:
- \( P \) is a finite set of places
- \( T \) is a finite set of transitions, \( P \cap T = \emptyset \)
- \( F \subseteq (P \times T) \cup (T \times P) \) is a set of arcs

This paper proposes a method based on Petri Net which can represent the implementation of each atomic process and crosscutting process. Petri Nets representing the implementation of process, atomic process and crosscutting process are named respectively Requirement Net, Process Net and Aspect Net. *Aspect net* includes Before-advice Net, After-advice Net and Around-advice Net. Each Requirement Net consists of place, transition, precondition, postcondition, input, output etc. The formal definition of Requirement Net is following.

**Definition of Requirement Net (RN)**: A Requirement Net is a 2-tuple \( RN = (SoP, SoA) \) where:
- \( SoP = (PN_1, PN_2, \ldots, PN_n) \) (n>0), it is a finite set of Process Nets.
- \( PN \) is a Petri Net.
- \( SoA = (AN_1, AN_2, \ldots, AN_n) \) (n>0), it is a finite set of Aspect Nets.
Definition of Aspect Net (AN): An Aspect Net is a 3-tuple AN=(SoAP, ANT, PN) where:
- SoAP=(PN1, PN2, …, PNN) (n>0), it is a finite set of atomic Process Nets.
- ANT∈{Before, After, Around}, it is the type of Aspect Nets.
- PN is a Petri Net.

As shown in Figure 5, Requirement Net meta-model is represented in UML.

![Figure 5. Requirement Net meta-model](image)

Requirement Net could be further refined and verified at design phase. For example, it can be verified that whether precondition and postcondition of Process Net are satisfied after weaving Aspect Net.

4. Case study

In this section, a case study for description of proposed method is offered. The case study is a simple ATM system of bank. Functions offered by ATM system are defined as follows.
- deposit funds
- withdraw funds
- query an account
- transfer funds
- Customer can delete a transaction during the operation above and ATM ejects card.
- Customer can access ATM system with high security.

We use GPRN framework for requirement modeling.

Goal model is shown in Figure 6. Deposit funds, Withdraw funds, Query an account and Transfer funds are functional requirement and Security is non-functional requirement. Delete a transaction could occur in all the functional requirements above and is a crosscutting goal.

![Figure 6. Goal model of ATM system](image)

To explain process modeling, Withdraw funds goal is taken for example. Process model of withdraw funds goal is shown in Figure 7. Validate PIN, Approving, Dispense cash and Eject card are executed in order. Validate PIN and Approving are composite processes but Dispense cash and Eject card are atomic processes. Control structure of validate PIN and Approving is sequence. Validate PIN consists of sequential execution of three atomic processes and Approving consists of sequential execution of two atomic processes. Dispense cash includes precondition and postcondition. Usertransaction is an atomic process and crosscuts Validate PIN, Approving and Dispense cash etc.
Implementation of each atomic process is represented by Requirement Net. To explain the modeling of Requirement Net, *Dispense cash* is taken for example.

Requirement Net for implementation of *Dispense cash* is shown in Figure 8. Requirement Net includes process net and aspect net which are used to represent the implementation of *Dispense cash* process and *UserTransaction* process respectively.

5. Conclusion

This paper presents a hierarchical GPRN framework for aspect-oriented requirement modeling. Requirements are decomposed by goal layer, process layer and requirement net layer. Meta-models of three layers are proposed and used to requirement modeling of aspect-oriented system. Requirement model with clear semantic of different layers interrelates each other. Compared with existing typical methods, GPRN framework thoroughly considers the characteristics of aspect-oriented requirements and hierarchically models *what*, *when*, and *how* in AORE respectively.
6. Acknowledgements

This research project was supported by the National Natural Science Foundation of China under Grant No. 60873024, Provincial Natural Science Foundation of Hubei under Grant No.2009CDB293.

7. References


