# An Aspect-Oriented Approach to Model Requirements

Lyrene Fernandes da Silva
*Computer Science Department – Catholic University of Rio de Janeiro - Brazil*
{lyrene,julio}@inf.puc-rio.br

## Abstract

*The principles of crosscutting concern separation and composition have been used by the Aspect-Oriented Development Community in order to solve the problems of tangling and scattering. In this work we present a proposal for integrating crosscutting concerns during the requirements engineering process. This approach uses goal models and the concepts defined in aspect-oriented languages to provide separation, composition and visualization of crosscutting concerns in order to facilitate their modeling and the traceability between them.*

## 1. Introduction

Requirements are continually changing and understanding their impact is a problem. This problem is even greater if we consider the crosscutting nature of requirements. Sometimes they influence or constrain each other, and this is known as crosscutting concerns [19].

The separation and composition of crosscutting concerns is a way of decreasing complexity and facilitating the analysis of each concern, both individually and in combination with others. These principles have been used in programming languages, by aspect-oriented languages [7]. However this level of abstraction hides many prior design decisions made without taking into account their crosscutting nature.

Research has been looking for higher abstractions related to aspects; modeling languages and methods have been proposed [4] and Requirement Approaches have mainly focused on the identification of candidate aspects [1]. In contrast to this, we are proposing a method for modeling requirements using concepts defined in aspect-oriented languages. This method involves separation, composition and visualization activities. We provide a modeling language based on goal models [13], a composition mechanism and different views of the created model. Our approach contributes to comprehension, evolution and reusability of requirement models.

The rest of this paper is organized as follows. In Section 2 we present the related work and main concepts used in our approach. In Section 3, we define a new approach to model requirements, the contributions and how we hope to validate this approach. In Section 4, we illustrate this approach with a case study. In the last Section, we present a summary and our conclusions.

## 2. Related work

This thesis is related to two main subjects: Requirement Modeling and Aspect-Oriented Development. We propose to use the concepts defined in aspect-oriented languages in order to reduce the difficulties related to the different characteristics of requirements and the problems in modeling and changing these requirements.

V-graph is the model used in our approach, which is a type of goal model [20]. Goal models represent the functional and non-functional requirements through decomposition trees [13]. V-graph is defined by goals, softgoals, tasks and the following decomposition relationships – contribution links (and, or, make, help, unknown, hurt, break) and correlation links (make, help, unknown, hurt, break). Each element has a Topic and a Type. The Type defines a generic functional or non-functional requirement. The Topic defines the context of that element.

V-graph was chosen because with this model we can consider requirements at three abstraction levels (softgoals, goals and tasks). This is important because in the same model we can represent reasons and operations, the context and how each element contributes to achieving the goals. Furthermore, there are important results in goal modeling concerning: how to analyze obstacles to the satisfaction of a goal [9]; how to qualitatively analyze the relationships in goal models; how to analyze variability [6]; how to analyze conflicts between goals through a propagation mechanism of labels [5]; how to identify aspects in goal models [20]; how to derive a feature model, a state model and a component model from goal models

[21]; and how to provide goal reuse [11] – this last work mentions a composition mechanism to integrate a goal model and a reusable goal model from a library.

Our method for separating and composing goal models does not change these approaches, but increases their potential. We have extended the goal models with information about how to compose them. We were influenced by aspect-oriented languages, which deal with crosscutting concerns in the implementation phase [7]. In AspectJ [8], for example, this separation is achieved by using a new element called 'aspect'. The combination is made by a component called 'weaver'. The 'weaver' processes the code, changing its elements, including the behavior or structure defined in the aspects. Similarly, we use the elements 'pointcut', 'advice' and 'intertype declaration' in order to represent how different goal models or parts of them affect each other.

It is not clear what an aspect is in the early stages of software production [19], but there are some approaches trying to provide techniques and methods for treating crosscutting concerns during the requirement process. Many of them aim to identify candidate aspects [1]. They use view points [14][15], lexical analysis [2] and catalogues of non-functional requirements [3][18]. Templates are used to describe how and where candidate aspects have an impact [12][3][15][18]. Use cases are used to represent functional requirements and the 'extend' relationship is used to represent candidate aspects [18]. Composition rules are also defined, but they are manually applied [18][2]. In Rashid's paper [15], an interesting way to automate this process using XML models is demonstrated. However, just one view is created from composition, and requirement sentences are used.

All these approaches differ from ours. First, we do not use the concept 'candidate aspect', because for us, knowing if a requirement will be an aspect in the implementation is not an issue at this point. We want to offer an easier way to model them. They may or may not be aspects in the code. The important thing is to be able to consider the scattering and tangling problems early on. Second, we want to model sets of requirements separately and offer a way to model the relationships between them. Furthermore, we want to offer different views originating from the composite model. Identifying crosscutting concerns is not our focus because they naturally appear during modeling. Crosscutting relationships are necessary, either because a requirement impacts on many points, or because it is important to keep one requirement separate from the others. Finally, we use goal models, which are an intentional view, and thus more informative representation than requirement sentences or use cases, and more user-friendly than templates.

## 3. Using aspects for facilitating the requirement modeling

*Lemma: Using concepts of aspect-oriented languages helps to deal with the tangling and scattering problems.*

*Hypothesis: Considering the tangling and scattering problems early on in the process improves the manageability of the software construction process.*

Some requirements scatter and tangle many others. This makes it difficult to modify the model and to perform impact analysis. In order to reduce these problems, we have defined an integration method for crosscutting concerns. The method is made up of three activities, called: separation, composition and visualization, see Figure 1.

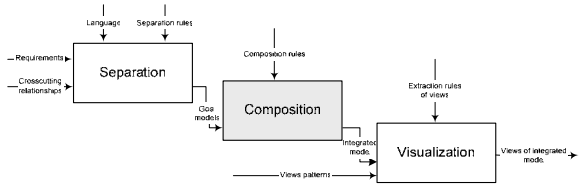


**Figure 1. Integration of crosscutting concerns**

The separation activity supports the requirement modeling - sets of requirements are modeled separately. In this way the complexity of modeling is reduced and the developer can consider each set of requirements more effectively. In order to model the requirements we developed a language based on the V-graph. This language is composed of a goal model specification and a crosscutting relationship specification [17]. Figure 4 shows information about the crosscutting relationship specification. We used the XML pattern to define the grammar of both models.

```
For each pointcut do {
  select advice
  for each operand do {
    if primitive = "add" then
      include advice as a sibling where operand_name = component_name
    if primitive = "include" then
      include advice as a child where operand_name = component_name
  ...
```

**Figure 2. Example of composition rules**

The composition activity achieves the combination of different goal models. This activity processes the crosscutting relationships creating a new goal model that contains all the original information. It uses composition rules, as shown in Figure 2. The composition activity is similar to the weaver in aspect-oriented languages. However, the weaver generates just one view of the system because computers are able to interpret (execute) complex models. In contrast to this, the visualization activity offers the developer different models or views [10]. This way, the

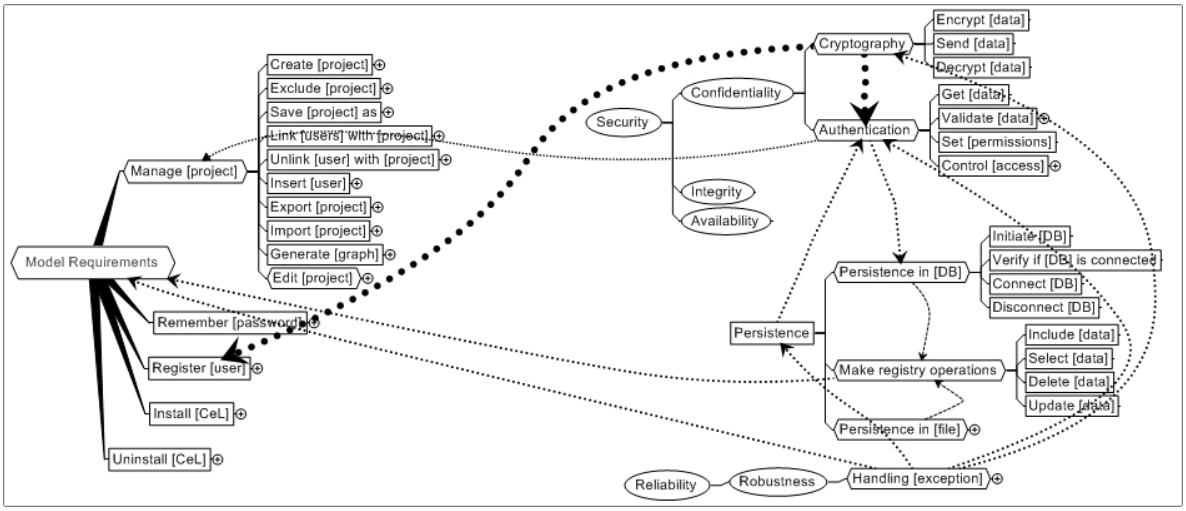


**Figure 3. Separation of goal models**

developer can continue elaborating the application model.

The idea is to provide requirement engineers with a way to model how the different concerns impact on each other. Therefore, while requirement engineers are modeling goal models they can concentrate on one group of requirements at a time and use crosscutting relationships to link these groups of requirements, representing the trace or impact between them. In order to be able to continue the modeling process, the engineer can obtain different views of the integrated model. This integrated model is created by an automatic composition mechanism.

### 3.1. Evaluation

In order to validate our approach, we are going to demonstrate our hypothesis through case studies. We will attempt to demonstrate that, through using some concepts of aspect-oriented languages for modeling requirements and providing views of the compound model, we will deal with the scattering and tangling problems during the requirement process. Therefore, we can consider, earlier on in the development process, some of the problems which may cause serious difficulties if they are only discovered during the implementation activity.

We are also implementing a set of tools to support our strategy. . Furthermore, we have modeled a set of crosscutting concerns that can be reused in different projects. Some examples are: Security, Persistence and Exception Handling. Although these examples are considered reusable, we know that each system may have a different definition for them. Therefore, our integration method helps the requirements engineer, facilitating the modeling of crosscutting concerns, the modification, and the analysis of these models.

## 4. Case study

This section presents an illustrative example of our approach. This example has four goal models: a goal model for an information system that helps to write scenarios and lexicon [16]; a goal model for Security; a goal model for Persistence; and a goal model for Reliability. Figure 3 shows these goal models. The ellipses are softgoals, hexagons are goals and rectangles are tasks. The pointed links are crosscutting relationships and the others are decomposition links.

Each crosscutting relationship has one or more pointcuts. Each pointcut is associated with ‘advices’ or ‘intertype declarations’. For example, the relationship between Cryptography and Authentication (in Figure 3) has two pointcuts, called encrypt and decrypt, see Figure 4. In this example each pointcut is associated to one advice. The advices define what from Cryptography model is going to be included into Authentication model.

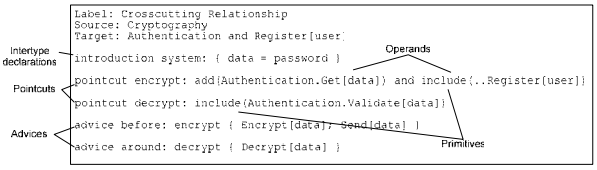


**Figure 4. Crosscutting relationship**

The crosscutting relationship links two elements in the same goal model and the composition mechanism processes this information creating a new goal model, see Figure 5 (one view of integrated model). In Figure 5 note the new decomposition relationships inserted into the original model. If these relationships are created manually, when changes occur, it is necessary to go through the whole model looking for where the change has had an effect. In contrast to this, in our method we can see how changes affect each part of

system separately. For example, if we decide that Cryptography in the system-to-be is unnecessary, we only have to eliminate the crosscutting relationships with Cryptography.

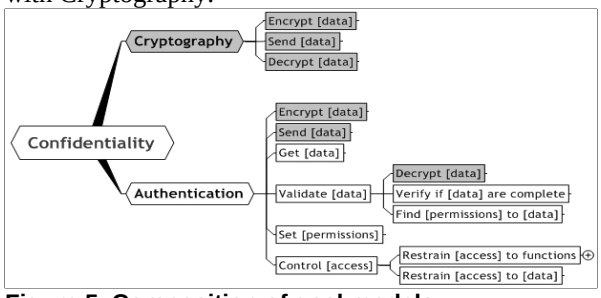


**Figure 5. Composition of goal models**

## 5. Conclusion

This thesis contributes mainly to modeling requirements, considering the tangling and scattering properties of functional and non-functional requirements. Our approach provides a new way to deal with crosscutting concerns early on in the development process. Using concepts of aspect-oriented languages, we have defined a method based on three main activities: separation, composition and visualization. "Separation" provides a language to model requirements, "composition" defines a component responsible for joining requirement models, and "visualization" makes it possible for the user to visualize different views of compound models.

Our approach improves the treatment of crosscutting concerns while defining requirements. We hope that it has a positive impact on the entire software development process. In order to implement this approach we are working on the development of a set of tools, the specification of a modeling language, the definition of composition rules, the definition of views to be extracted from goal models and on the modeling of a set of reusable crosscutting requirements.

## 6. References


[1] J. Bakker, B. Tekinerdogan and M. Aksit, "Characterization of early aspects approaches", Proceedings of the Early Aspects Workshop at AOSD, 2005.

[2] E. Baniassad and S. Clarke, "Theme: An approach for aspect-oriented analysis and design", 26th International Conference on Software Engineering (ICSE'04), Scotland, 2004, pp. 158-167.

[3] I. Brito and A. Moreira, "Integrating the NFR framework in an RE model", Proceedings of the Early Aspects Workshop at AOSD, England, 2004.

[4] C. Chavez, "A Model-Driven approach to aspect-oriented design", PhD Thesis, Computer Science Department, PUC-Rio, Rio de Janeiro, Brazil, 2004.

[5] P. Giorgini, J. Mylopoulos, E. Nicchiarelli and R. Sebastián, "Reasoning with goal models", Proceedings of the 21st International Conference on Conceptual Modeling, 2002, pp. 167-181.

[6] B. Gonzáles, M. Laguna and J. Leite, "Visual variability analysis with goal models", Proceedings of IEEE International Symposium on Requirements Engineering (RE'04), Japan, 2004, pp. 38-47.

[7] G. Kiczales et al., "Aspect-oriented programming", Proceedings of the European Conference on Object-Oriented Programming (ECOOP'97), LNCS (1241), Springer-Verlag, Finland, 1997.

[8] G. Kiczales et al., "An overview of aspectJ", Proceedings of the European Conference on Object-Oriented Programming (ECOOP'01), Hungary, 2001.

[9] A. Lamsweerde and E. Letier, "Handling obstacles in goal-oriented requirements engineering", IEEE Transaction Software Engineering, 26(10):978–1005, 2000.

[10] J. Leite and P. Freeman, "Requirements Validation Through Viewpoint Resolution", IEEE Transactions on Software Engineering: Vol. 17, N. 12, 1991, pp 1253-1269.

[11] J. Leite, Y. Yu, L. Liu, E. Yu and J. Mylopoulos, "Quality-Based Software Reuse", Proceedings of the CAiSE 2005-LNCS 3520, 2005, pp. 535-550.

[12] A. Moreira, J. Araújo and I. Brito, "Crosscutting quality attributes for requirements engineering", Proceeding of the 14th International Conference on Software Engineering and Knowledge Engineering, ACM Press, Italy, 2002.

[13] J. Mylopoulos, L. Chung, and B. Nixon, "Representing and using nonfunctional requirements: A process-oriented approach", IEEE Transactions on Software Engineering, 18(6):483–497, June 1992.

[14] A. Rashid, P. Sawyer, A. Moreira and J. Araújo, "Early aspects: a model for aspect-oriented requirements engineering", Proceedings of IEEE Joint Conference on Requirements Engineering, Germany, 2002, pp. 199-202.

[15] A. Rashid, A. Moreira and J. Araújo, "Modularization and composition of aspectual requirements", Proceedings of the 2nd International Conference on Aspect-Oriented Software Development, ACM, 2003, pp. 11-20.

[16] L. Silva, J. Leite and K. Breitman, "Teaching Software Engineering: Report on Experiments", Education in Computing Workshop (XII WEI ), 2004 (in Portuguese).

[17] L. Silva, J. Leite, "An aspect-oriented language for requirement modeling", Proceedings of the Requirement Engineering Workshop at CAiSE 2005, Porto-Portugal, 2005, pp. 13-25. (in Portuguese)

[18] G. Sousa, S. Soares, P. Borba and J. Castro, "Separation of crosscutting concerns from requirements to design: Adapting the use case driven approach", Proceedings of the Early Aspects Workshop at AOSD, England, 2004.

[19] B. Tekinerdoðan, A. Moreira, J. Araújo, P. Clements, "Early aspects: aspect-oriented requirements engineering and architecture design", Report on Early Aspects Workshop at AOSD, England, 2004.

[20] Y. Yu, J. Leite and J. Mylopoulos, "From goals to aspects: discovering aspects from requirements goal models", Proceedings of IEEE International Symposium on Requirements Engineering (RE'04), Japan, 2004, pp. 38-47.

[21] Y. Yu, J. Mylopolous, A. Lapouchnian, S. Liaskos and J. Leite, "From stakeholder goals to high-variability software design", Internal report, 2005.