

Addressing the Separation of Concerns by means of an Automatic Refactoring Process

Santiago A. Vidal^{1, 2}, Claudia A. Marcos¹

¹ISISTAN Research Institute, Faculty of Sciences, UNICEN University
Campus Universitario, Pje. Arroyo Seco (B7001BBO)
Tandil, Buenos Aires, Argentina

²CONICET (National Council for Scientific and Technological Research), Argentina
{svidal, cmarcos}@exa.unicen.edu.ar

Aspect-Oriented Software Development (AOSD) encapsulates the crosscutting concerns (CCCs) increasing the software modularization and reducing the impact when changes are made in the code thereby improving the systems adaptability and reusability.

For existing object-oriented (OO) systems to incorporate the benefits of AOSD, those systems are usually re-modularized into aspect-oriented (AO) systems. This leads to a need for techniques and tools that can help developers with the identification of crosscutting concerns, called aspect mining, and then with the refactoring of those concerns into aspects, called aspect refactoring.

We think that the migration from an OO system to an AO one improves the structure and quality of the software, and thus eases software evolution. Along this line, we believe that the provision of semi-automated support to help the developer to discover crosscutting concerns in legacy systems and to encapsulate them into aspects is really beneficial. For this reason, we focus on the activity of aspect refactoring proposing an iterative process that helps developers to achieve the task of migrating an object-oriented system into an aspect-oriented one by analyzing and applying aspect refactorings. So, by means of this process we expect to improve the maintenance of software systems through the modularization of system functional concerns and crosscutting concerns, and achieving a better flexibility and extensibility of the resultant code. In order to provide automated support, the proposed process uses artificial intelligence techniques. Specifically, we want to automatically identify which aspect refactoring (or a set of them) must be applied given a specific fragment of aspectizable code. Also, we want to predict the order in which the crosscutting concern should be refactorized and additional activities to the refactoring that must be done. These automatic approaches are based on association rules algorithms and hidden Markov models.

Currently, the mechanisms for identification based on association rules and the mechanisms for identification based on hidden Markov models have already been developed. Also, the first results of this project have been published conducting case studies in real system. However, more experimentation in real system will be performed with the goal of evaluating the approach.