Mining Early Aspects based on Syntactical and Dependency Analyses[*†]

José M. Conejero[1], Juan Hernández[1], Elena Jurado[1] and Klaas van den Berg[2]

[1] Quercus Software Engineering Group, Universidad de Extremadura,
Avda. de la Universidad s/n, 10071 Cáceres, Spain
{chemacm, juanher, elenajur}@unex.es
[2] Software Engineering Group, University of Twente,
7500 AE Enschede, the Netherlands
k.g.vandenberg@ewi.utwente.nl

Summary

There are different approaches in the requirements engineering community to deal with crosscutting concerns, introducing the benefits of the application of aspect-orientation at these early stages of development. However, these approaches lack a formal definition of crosscutting to be based on. Most of these approaches rely on the use of Natural Language Processing techniques for aspect identification in textual documents and thus, they lack a unified process that generalises its application to other requirements artefacts such as use case diagrams or viewpoints. In this paper, we propose a process for mining early aspects, i.e. identifying crosscutting concerns at the requirements level.

First, the aspect mining process is based on a conceptual framework that is independent of specific requirements artefacts and that provides a formal definition of crosscutting based on the trace relations or mappings that exist between two different domains, source and target. The aspect mining process relies on syntactical and dependency analyses at the requirements level that allow the process to be automated since the existing mappings between the two domains are automatically obtained. Second, early aspect refactoring is given for UML use cases diagrams. The refactoring process is driven by a set of concern-oriented metrics that provide modularity assessments (i.e. quantification of the degree of scattering, tangling and crosscutting). Third, our process is both validated by the utilisation of the concern metrics at requirements level and compared with other early aspect mining proposals.

The aspect mining process. In this paper, a specific instantiation of the process to be applied at the requirements level is presented. The process has been applied using concerns and use case diagrams as source and target domains, respectively. Thus, the aspect mining process aims at identifying crosscutting situations based on concerns scattered over different use cases and where other concerns are tangled.

---

[*] Conejero, J., Hernández, J., Jurado, E. and Berg, K. van den. Mining early aspects based on syntactical and dependency analyses. Elsevier Science of Computer Programming vol. 75, Issue 11, 1 November 2010, 1113--1141 (2010)

[†] This work has been carried out with the support by MEC under contract TIN2008-02985

The main steps of our approach are outlined in Fig. 1 and summarised as follows:

(A) Identifying source elements. We analyse the system requirements to identify the main concerns: functional and non-functional. To identify the non-functional concerns (NFCs), a catalogue is used where common NFCs are defined. Both functional and NFCs are represented in XML format.

(B) Identifying target elements. Requirements are modelled using use cases. As with concerns, requirements are also represented in an XML format, exporting the use case diagrams to XMI.

(C) Build dependency matrix. Taking concerns and requirements as source and target respectively we establish the trace relations between them. These relations are automatically obtained by syntactical and dependenciy analyses.

(D) Identification of crosscutting by matrix operations. The next step consists of the application of several simple matrix operations to obtain the crosscutting concerns at the requirements level.

(E) Empirical analysis. Based on the matrices obtained in the previous step, our concern-metrics are calculated. Based on the values obtained for the different metrics, the source and target elements with a higher degree of crosscutting and tangling are identified, respectively.

(F) Aspect-oriented refactoring. Finally, the crosscutting concerns identified are modelled using aspect-oriented techniques improving, thus, modularity and the reusability of the system.

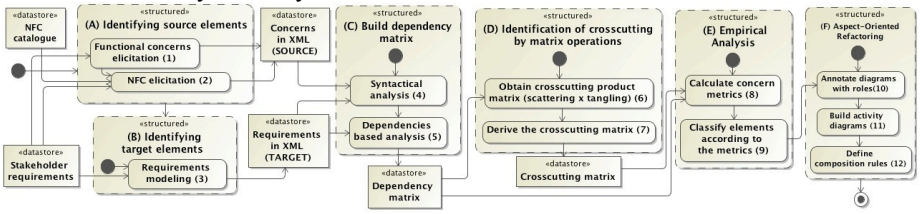

Fig. 1. Main phases of the aspect mining process

The validation. A double validation of the process is presented. Firstly, the application of the process to a real system is illustrated. We apply the set of concern-oriented metrics presented and compare the results obtained by our metrics with those provided by other authors' similar metrics. We show that our metrics obtain values consistent with the other metrics, validating our aspect mining process. The results provided by the metrics show that the NFCs are those with a higher degree of crosscutting. The process also provides interesting results regarding functional concerns that also present a high degree of crosscutting being, thus, candidate to be refactored. The comparison with similar metrics comes to confirm the results obtained by our process and the metrics applied.

Secondly, we compare the framework with similar early aspect approaches, such as EA-Miner or Theme/DOC. We discuss some interesting open issues that may be considered regarding the aspect mining process presented. The comparison shows that our process obtains similar results to those obtained by similar semi-automatic approaches. It also illustrates how manual approaches could obtain better results in small examples but their application to real systems could be unfeasible making, thus, the (semi)automatization of the process needed.