

# Early-AIM: An Approach for Identifying Aspects in Requirements

Américo Sampaio, Awais Rashid and Paul Rayson

Computing Department, Lancaster University, Lancaster, UK  
{a.sampaio, marash, paul}@comp.lancs.ac.uk

## Abstract

*Identifying aspects at an early stage helps to achieve separation of crosscutting concerns in the initial system analysis, instead of deferring such decisions to later stages of design and code, and thus, having to perform costly refactorings. This paper describes the Early-AIM approach that utilises corpus-based natural language processing (NLP) techniques to effectively enable the identification and modelling of early aspects in a semi-automated way.*

## 1. Introduction

Aspect-Oriented Software Development (AOSD) is a technique that has been showing encouraging results in improving separation of concerns, and achieving better software modularization and maintainability. In order to maximize its benefits, AOSD should be used from the early stages of software development such as domain analysis and requirements engineering [1- 4]. Identifying aspects at an early stage helps to achieve separation of concerns in the initial system analysis, instead of deferring such decisions to later stages of design and code, and thus, having to perform costly refactorings.

An important step towards effectively supporting early aspects identification is to provide not only abstractions that represent crosscutting requirements, but also to offer mechanisms e.g., tool support for mining aspects in requirements documents [1, 4]. This helps the requirements engineer to identify early aspects and their relationships with other requirements.

Current techniques for identifying aspects in requirements are ineffective when requirements are complex or unstructured. These techniques rely upon limited mechanisms for aspect mining, which impose a specific structure on requirements documents, e.g., [1], request large inputs from the user e.g., [1], or do not offer semantic reasoning about what is being considered an aspect e.g., [1, 4].

This paper describes an approach that utilises corpus-based

natural language processing (NLP) [8] techniques to effectively enable the identification of early aspects in a semi-automated way. The technique proposed here describes how different sources of requirements (e.g., interviews, natural language descriptions of the system) can be automatically mined to help the requirements engineer quickly identify and build a structured aspect-oriented model of the system's requirements.

The main goal of Early Aspects Identification Method (Early-AIM) is to determine potential aspect candidates in requirements documents regardless of how they are structured. The approach uses NLP techniques which provide support for context sensitive analysis of requirements [5, 7]. Tool support (EA-Miner tool) is provided to help the developer automatically mine and model the crosscutting concerns without having to previously read the requirements documents. The next section presents an overview of the approach.

## 2. Early-AIM (Early Aspects Identification Method)

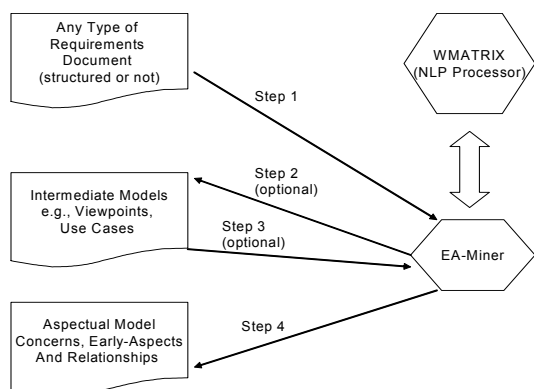
The Early-AIM approach utilises the WMATRIX [5] NLP Processor that provides features such as part-of-speech and semantic tagging, frequency analysis and concordances. This processor is capable of tagging any type of text written in English (e.g., a newspaper text, a Shakespeare text, a requirements document, etc.).

As natural language is a convenient way of expressing requirements, research in [5] and [7] has focused on using WMATRIX features to help in the automation of the identification of certain concepts such as objects, user roles, functionalities and early aspects in requirements documents. In our case, the EA-Miner tool utilises WMATRIX for the specific purpose of identifying early aspects and relationships between requirements. The approach is shown in Figure 1 and summarised next. For more details on the approach see [7].

The approach begins in Step 1 by analysing existing documents that are sources for requirements elicitation such as interviews done with stakeholders (e.g., clients, managers, users) or informal descriptions of the system. The EA-Miner

tool reads these files and passes them to WMATRIX to be processed.

The next two steps (2 and 3) are optional and aim at facilitating the production of the aspectual model in Step 4 by producing a more structured model of the requirements for input. EA-Miner helps to automatically identify elements such as viewpoints and use cases based on some criteria such as identifying verbs that denote action (e.g., sell, buy, insert) as possible use cases or nouns (e.g., client, driver, ATM machine) as possible viewpoints.



**Figure 1 – Early-AIM Method**

In Step 4 a model is produced showing the concerns, early aspects and their relationships. The implementation of this step is based on the following issues:

- *Action words*: the tool can automatically look for action words, identified as verbs by the WMATRIX tool, and produce a model that represents the relationships between the requirements based on statistical data such as collocations to reason about what action words affect each other. Different than Theme/Doc [1], the engineer does not have to provide the action words in advance and, also, the natural language processor enables a more context sensitive analysis of the words recognising for example, that the actions “collect” and “pick up” are in the same semantic field.
- *Semantic analysis and domain lexicon*: a domain specific lexicon, based on the NFR [6] catalogues, categorises well known crosscutting concerns. EA-Miner can then parse the text and tag the possible early aspects helped by previous semantic tagging (done by WMATRIX) and by the domain lexicon to avoid choosing erroneous candidates.
- *Meta-models and heuristics*: to produce the output of Step 4 (Aspectual Model) EA-Miner looks for the identified candidate aspects and also for intermediate models specific input such as their meta-models and construction heuristics.

It is important to point out that our approach for mining aspects can be used regardless of the structure of the textual document provided as input (e.g., informal descriptions, interviews, structured documents). Document-heavy domains based on regulations, standards and various types of extensive documentation can thus benefit from our approach that helps to automate the requirements engineering process.

Regarding the scalability of the approach, data found in [5]

shows that WMATRIX tagging process is not a bottleneck since the tool has shown good results (time varies linearly with the document size) tagging several large documents (tens of thousands of words). Moreover, the tool offers a very good precision on its results [5, 9] 98% and 91% for, respectively, syntactic and semantic tagging.

### 3. Future Work

Our future work will focus on continuing the implementation of EA-Miner and validating it in some case studies. The tool will provide support for automating the approach presented in this paper by using NLP features provided in the WMATRIX tool such as syntactic and semantic tagging and statistical data such as frequency of words and collocations.

Our focus now is to investigate how we can tailor the process of tagging the documents to reflect domain specific contextualization. For this purpose we are planning to create a domain specific lexicon and parser for some kinds of known crosscutting concerns based on the NFR [6] catalogues and some rules of disambiguation, which can be aided by the previous tagging provided by WMATRIX. Moreover, we intend to make the approach work with different intermediate models (e.g., viewpoints, use cases) by providing a way to input their meta-models and also heuristic rules for interpreting them and producing the specific aspectual model.

**Acknowledgement:** This work is supported by European Commission grant IST-2-004349: European Network of Excellence on Aspect-Oriented Software Development (AOSD-Europe), 2004-2008.

### References

- [1] E. Baniassad and S. Clarke. “Finding Aspects in Requirements with Theme/Doc.” In Proceedings of Early Aspects 2004: (AOSD 2004) Lancaster, UK, 22 Mar. 2004.
- [2] A. Rashid, A. Moreira, and J. Araujo. “Modularisation and Composition of Aspectual Requirements.” In Proceedings of the 2nd International Conference on Aspect-Oriented Software Development, pages 11–20. ACM Press, 2003.
- [3] A. Rashid, P. Sawyer, A. Moreira, and J. Araujo. “Early Aspects: A Model for Aspect-Oriented Requirements Engineering.” In Proceedings of IEEE Joint International Conference on Requirements Engineering (RE 2002), pages 199–202. IEEE Computer Society, 2002.
- [4] L. Rosenhainer. “Identifying Crosscutting Concerns in Requirements Specifications.” In Proceedings of OOPSLA Early Aspects 2004: Aspect-Oriented Requirements Engineering and Architecture Design Workshop. October 2004, Vancouver, Canada.
- [5] P. Sawyer, P. Rayson, and R. Garside. “REVERE: support for requirements synthesis from documents.” Information Systems Frontiers Journal. Volume 4, issue 3, Kluwer, Netherlands, pp. 343 – 353, 2000.
- [6] L. Chung et al. “Non-Functional Requirements in Software Engineering.” Kluwer Academic Publishers, 2000.
- [7] A. Sampaio et al. “Mining Aspects in Requirements.” presented at Early Aspects 2005 (AOSD 2005), Chicago, Illinois, USA, 2005.
- [8] R. Garside et al. “Corpus Annotation: Linguistic Information from Computer Text Corpora.” London & New York: Longman, 1997.
- [9] Rayson, P., Archer, D., Piao, S. L., McEnery, T. (2004). The UCREL semantic analysis system. In proceedings of the workshop on Beyond Named Entity Recognition Semantic labelling for NLP tasks in association with (LREC 2004), 25th May 2004, Lisbon, Portugal.