A Systematic Review on Mining Techniques for Crosscutting Concerns

Rafael Serapilha Durelli[†], Daniel San Martín Santibáñez[*], Márcio Eduardo Delamaro[†] and Valter Vieira de Camargo[*]
[*]Departamento de Computação, Universidade Federal de São Carlos,
Caixa Postal 676 – 13.565-905, São Carlos – SP – Brazil
Email: {daniel.santibanez,valter}@dc.ufscar.br
[†]Instituto de Ciências Matemáticas e Computação, Universidade de São Paulo,
Av. Trabalhador São Carlense, 400, São Carlos – SP – Brazil
Email: {rdurelli, delamaro}@icmc.usp.br

Abstract—**Background: The several maintenance tasks a system is submitted during its life usually causes its architecture deviates from the original conceivable design, ending up with scattered and tangled concerns across the software. Mining techniques for crosscutting concern attempt to identify such scattered and tangled concerns in a program to support maintenance, reverse-engineering or the first step of a reestructuration process towards an aspect-oriented software. Objectives: Conducting a systematic review aiming at identifying the main tools and techniques employed in this process and also verify the existence of combined mining approaches. Methods: We have carried out a systematic review based on searching of major electronic databases. Results: We selected and classified 62 papers by their mining technique, type and year of publication. Among these papers were identified 18 mining techniques for crosscutting concern. Furthermore, we believe strongly that the results of this review may serve as a roadmap to potential users of this type of techniques, e.g., helping future researchers in selecting an appropriate technique. Conclusions: Our review shows that in the literature there is a considerable number of mining techniques for crosscutting concern. It is interesting to note that a large amount of this techniques are intended to identify several types of crosscutting concerns but there is a lack of research to say which technique is most suitable for identifying a specific crosscutting concern. Also, we believe it is necessary to perform studies showing the pros and cons of using combinations of several mining techniques for crosscutting concern in a unique software environment.**

I. Introduction

Mining techniques for crosscutting concern is indispensable for software maintenance, reverse engineering, re-engineering and even for re-documentation. However, manually applying mining techniques for crosscutting concern is difficult and error-prone [1], once, this sort of system tends to (*i*) have complex architectures with several clones spread out through the source code; (*ii*) involve several kinds of crosscutting concerns, such as patterns, architectural styles, business rules and non-functional properties; (*iii*) be very large, making the manual mining impractical. Thus, there is a need to use techniques and automated or semi-automated tools, which can aid software engineers to locate crosscutting concern into legacy systems. In this context, the research area which aims to investigate techniques and tools to increase the mining of crosscutting concerns is known as "concern mining" [2].

In this paper we present a systematic review with the aim of identifying a large amount of techniques related to crosscutting concern mining. As a consequence, we have discovered that in the last years a considerable number of mining techniques for crosscutting concern are available within academic context. Furthermore, this systematic review shows that there are a lot of interesting and important research topics that could be investigated yet.

II. The Systematic Review

This study has been undertaken as a systematic review based on the guidelines proposed by Kitchenham and Brereton [3]. According to them, in order to conduct a systematic review it is advisable to follow three main phases: (*i*) planning the review – (*ii*) conduction the review – and (*iii*) reporting the review. Following sections present slightly details on how each phase was carried out.

A. Planning the Systematic Review

In this phase we have defined the review protocol. This protocol contains: (*i*) the research questions, (*ii*) the search strategy, (*iii*) the inclusion and exclusion criteria and the (*iv*) data extraction and synthesis method. In this setting, three research questions were devised, they are:

- **RQ$_1$:** Which techniques for mining crosscutting concerns are used within academic contexts? Furthermore, which ones are more and less explored?
- **RQ$_2$:** Which evaluation techniques (e.g., case study, empirical strategies, comparative experiments) have been employed to evaluate these techniques?
- **RQ$_3$:** Considering the techniques that we found, which ones have automated support?

Afterwards, we have defined the search string and the electronic databases. Figure 1 shows the search string which we have used. We have used the search string on the following electronic databases: *ACM* (www.portal.acm.org), *IEEE* (www.ieeexplore.ieee.org), *Scopus* (www.scopus.com) and *Springer* (www.springer.com/lncs). No limits were placed on date of publication with a view to not restrict the review study scope.

*(("approach") **OR** ("method") **OR** ("technique") **OR** ("methodology")) **AND** (("aspect oriented") **OR** ("aspect-oriented")) **AND** (("aspect mining") **OR** ("concern mining") **OR** ("coding mining") **OR** ("code mining")) **AND** (("crosscutting concerns") **OR** ("cross-cutting concerns") **OR** ("Separation of Concern") **OR** ("SoC")) **AND NOT** (("data mining") **OR** ("early aspects") **OR** ("product lines") **OR** ("mining of web"))*

Fig. 1. Search String.

The Inclusion criteria devised and applied are: **IC_1:** the primary study presents at least one mining technique for crosscutting concern and **IC_2:** the primary study presents at least one type of evaluation technique for crosscutting concern. Similarly, the exclusion criteria are: **EC_1:** the primary study is not about mining techniques for crosscutting concern – **EC_2:** the primary study presents data mining technique, however, such technique is applied to databases and not for crosscutting concern mining – **EC_3:** the primary study is a short paper (papers with twos pages or less).

B. Conducting the Systematic Review

The database Scopus has returned more primary studies than the others databases (262), i.e., IEEE, ACM and Springer have returned 215, 202 and 127, respectively. Possibly, this came about because this database indexes studies of others databases, such as IEEE and Springer. Summing up, we have gotten 806 primary studies. After, we read the the titles and abstracts and applied the inclusion and exclusion criteria. As a result, we have gotten a total of 124 primary studies that were read entirely, so the upshot obtained were 62.

C. Reporting the Systematic Review

Aiming to answer the RQ_1 we have read the 62 primary studies, and thereby we have identified 18 techniques of crosscutting concern mining. Therefore, each included primary study was assigned to one or more techniques. Additionally, we classify those techniques in two major categories: (*i*) Generative approach and (*ii*) Query-based. The first ones aims at extracting and generating aspect seeds automatically using structural information of the program source code while the second ones utilizes manual input such as textual pattern.

Figure 2 illustrates the frequency of primary studies related to techniques which were found. It may be seem clearly that Fan-In Analysis followed by Identifier Analysis are the more explored. Similarly, Program Analysis Based, XScan-Conern-Peers, Data-Flow Concern Identification and Model-Driven are less explored.

Aiming to answer the RQ_2 we have analyzed individually the 62 primary studies focus on gather which empirical strategies them have employed to validate the crosscutting concern mining techniques. Thus, among the 62 primary studies, 28 have been carried out experiments to validate their crosscutting concern mining techniques. Similarly, 24 primary studies have been employed some sort of case studies, and 17 neither have carried out experiments nor case studies to validate their techniques for mining crosscutting concern.

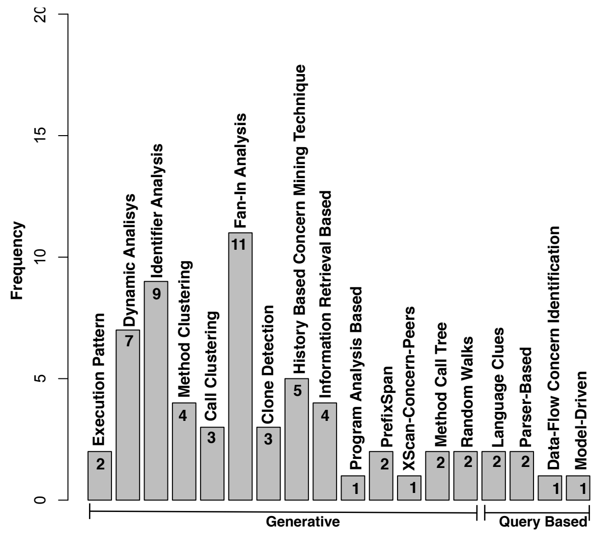

Fig. 2. Frequency of studies in each category.

As for answering the RQ_3 we have read the 62 primary studies, as consequence, we found 33 tools. They are: DynAMiT, Dynamo, CERBERUS, AspectRT, DelfStof, Find-Concept tool, Lexical-Chaining tool, ConAn, AMAV, SOM, CBFA, FAN-in metric tool, Fint, CBFA, PDG-DUP, ccdiml, CCFinder, COMMIT, HAM, line co-change, tool-chain, MSAM, Aspect Browser, Grep, JQuery, FEAT, DRACO-PUC, AMT, FUNG, Xscan, RRAM, CoDEx and Prism CC.

III. Concluding Remarks

The main contribution of this paper is to provide a basic knowledge and state of art of several techniques found in the literature. It can be used to introduce new researchers in concern mining research area.

Although there are some tools which combine concern mining techniques, it is important to perform studies showing the pros and cons of using combinations of several mining techniques for crosscutting concern in a unique software environment. Another interesting issue is to know which technique is most suitable for identifying a specific crosscutting concern.

Acknowledgment

The authors would like to thanks the financial support provide by CAPES. Rafael S. Durelli also would like to thanks the financial support provided by FAPESP (Process 2012/05168-4).

References

[1] M. Eaddy, A. Aho, and G. C. Murphy, "Identifying, assigning, and quantifying crosscutting concerns," in *Proceedings of the First International Workshop on Assessment of Contemporary Modularization Techniques*. Washington, DC, USA: IEEE Computer Society, 2007.

[2] B. Adams, Z. M. Jiang, and A. E. Hassan, "Identifying crosscutting concerns using historical code changes," in *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 1*, ser. ICSE 10. New York, NY, USA: ACM, 2010, pp. 305–314.

[3] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering - a systematic literature review," *Information and Software Technology*, vol. 51, pp. 7–15, January 2009.