A Systematic Literature Review on Aspect Mining for Migration of Legacy Systems to Aspect-Oriented Systems

Noopur Goel
Assistant Professor, Department of Computer Applications, VBS Purvanchal University, Jaunpur, U.P., India

Abstract

Organizations depend heavily on their business-criticalsoftware systems that have been developed before three decades or more and thus became legacy system. Evolution and maintainability of legacy systems is all time attention drawing subject for researchers and especially practitioners. Discovering the crosscutting concerns and separating it from core functionalities of a software system may help in evolution of the legacy systems. Aspect-oriented software development (AOSD) tries to achieve the goal. AOSD is new programming paradigm which helps to bring in modularity in the program by writing the crosscutting concerns in the form of 'aspects'. Modularity brings comprehensibility and hence maintainability of the software system. Tools and techniques, which aid in identifying the crosscutting concerns in such systems and refactoring them into aspects, are needed to apply aspect-oriented techniques to legacy systems at use in industry.

There has been a significant development in the migration of legacy to Aspect oriented systems. This paper aims to identify which approach is mostly used in the migration from legacy systems to aspect-oriented software systems. A systematic literature review is conducted which identified 48research papers on various aspect mining approaches, techniques and tools proposed and published by the researchersfrom1998 to 2014. For this purpose, 126 primary studies were collected. The resulting evaluation forms the database of current research approaches, methods and techniques used in migrating Legacy systems to Aspect-oriented systems.

Introduction

Legacy systems in business industries are very large and complex systems. Evolution of the software systems is inherent due to many causes. A system decomposed into a well modularized system i.e. functions and classes may have some functionality that cut across that modularity. This is often referred to as *tyranny of the dominant decomposition* (Tarr, Ossher, Harrison, & Sutton Jr, 1999) and such functionalities are called crosscutting concerns because they are spread over many decomposition units. Examples of crosscutting concerns are logging, synchronization, exception handling, persistence, exception handling, and error management. Many crosscutting concerns are spread, either scattered or tangled, all over the code. This leads to the problem of maintenance and understandability of software systems. Identification and modularization of these crosscutting concerns are very difficult. Aspects-oriented techniques can be applied to the legacy systems in the business industry, i.e. there is a need to migrate the legacy codes into the aspect-oriented systems. Aspects represent the non-functional requirements or behaviors of the system. They are the non-functional requirements or –ilities of the system. In order to transform the legacy systems to aspect-oriented systems, there is a need of tools and techniques that can help in identifying the crosscutting concerns in the systems and refactoring them into aspects. Migration of the legacy codes into the aspect-oriented systems is composed of aspect mining and aspect refactoring.
Aspect Mining is a reverse engineering process of identifying the crosscutting concerns in the given source code of the legacy system that can be

potentially converted into aspects. Such concerns are referred to as 'aspects candidates'.
Aspect Refactoring is the process of converting the identified aspect candidates into real aspects in the source code.

Organizations depend heavily on their business-critical software systems that have been developed before three decades or more and thus became legacy system. Evolution and maintainability of legacy systems is all time attention drawing subject for researchers and especially practitioners. Discovering the crosscutting concerns and separating it from core functionalities of a software system may help in evolution of the legacy systems. Aspect-oriented software development (AOSD) tries to achieve the goal. AOSD is new programming paradigm which helps to bring in modularity in the program by writing the crosscutting concerns in the form of 'aspects'. Modularity brings comprehensibility and hence maintainability of the software system.

In this paper, a systematic literature review(SLR) of the existing literature of aspects or crosscutting concerns mining is performed. This paper focuses on identifying techniques, methods and approaches that are relevant to the migration of legacy to aspect-orientedsystems. The maincontribution of this research isthat it provides an inventory of methods and techniques used in various phases of aspect-mining.This database will benefit researchers and industrial practitioners so that the researchers can contribute to promote the legacy to aspect-oriented systems, whereas theindustrial practitioners can implementvarious methods and techniques that are stated in research in realworld practices.

Research Method

The systematic review processis conducted on the basis of guidelinesproposed by Kitchenham (Kitchenham, 2007).

A systematic literature review(often referred to as a systematic review) is a means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest. Individual studies contributing to a systematic review are called *primary* studies; a systematic review is a form of *secondary* study [Kitchenham,2007].

Such a review process is suitable in this research work since it summarizes the existing contributions, and provides a background to place new research activities in a research framework. Various stages of systematic literature review used in this paper are as follows:

i. Planning the Review
   Identification of the need for a review
   Specifying the research question(s)
   Developing a review protocol
ii. Conducting the Review
iii. Reporting the Review.

Planning the Review

Before proceeding for a systematic review, it is required to validate the need for such a review. *Defining the research questions(s)* which the systematic review will address and *producing a review protocol* (i.e. plan) which defines the basic review procedures are the most important pre-review activities.

Identification of the need for a review

Evolution of the software systems is inherent due to many causes (Khadka*et al.*, 2014, Iyer, 2008): *To Remain Agile to Change*, *High Cost of Maintenance*, *Lack of Knowledge/Resources and Prone to Failure.* A number of benefits of modernizing legacy system are observed by the practitioners in the industry (Khadka*et al.*, 2014): *Business Critical, Proven Technology, Reliable Systems and performance*. Due to the large and complex structure of legacy system and spreading of cross-cutting concerns all over the code, problem of maintenance and understandability of

software systems arises. Mining techniques for crosscutting concerns are crucial for *software maintenance, reverse engineering, reengineering and even for re-documentation*. On the other hand, manual identification and modularization of the cross-cutting concerns from such a legacy systemare very difficult and error-prone.As a consequence, a number of approaches, tools and techniques have been developedand are used in the industry, which aid in identifying the crosscutting concerns in such legacy systems and are converted to 'aspects' after applying refactoring.

This study is motivated by the problem (Yuen, &Robillard, 2007): How to identify crosscutting concerns in a legacy system, i.e. to select a single or combination of two or more adequate aspect mining technique.

As a part of the present work, it was found that a SLR by Durelli*et al.* [2] is performed, which focused on three things: first- identification of techniques employed in mining cross-cutting concerns, second- the extension of taxonomy presented by Kellens et al. and third- the possible combination of these tools and techniques that improve recall and precision metrics for Persistence and Observer concerns. In this paper, a systematic literature review (SLR) of the existing approaches, techniques and tools of crosscutting concerns mining in legacy systems is performed.

Specifying the research question(s)

A set of research questions is needed in order to focus on the review theme. The core idea of this systematic literature review is to make an overview of legacy to AOP evolution approaches. In order to achieve the objective, research questions are formulated in the subsection 2.1.3.2.

Developing a review protocol

A review protocol is an action plan whichwill be used to carry out a specific systematic review. The components of a protocol include *research question, data sources, search strategy, study selection strategy, data extraction,* and*data synthesis.* The first four define the scope and motivation of the research while the last two describehow the results are concluded from the data.

Research Questions

A set of research questions is needed in order to focus on the review theme. The core idea of this systematic literature review is to make an overview of legacy to AOP evolution approaches. In order to achieve the objective, following research questions are formulated.

RQ1:*Which approach is mostly used in the migration from legacy systems to aspect-oriented software system?*

RQ2:*In which yearmost of the research papers are published regarding the migration from legacy systems to aspect-oriented software system?*

Data Sources

The resources, digital libraries/journals, searched for the primary study include: ACM Digital Library ACM (www.portal.acm.org), IEEE (www.ieeexplore.ieee.org), Scopus (www.scopus.com) and Springer (www.springer.com/lncs), SpringerLink, CiteseerX (www.citeceerX.org) , ScienceDirect (www.sciencedirect.com), and Wiley Inter Science Journal Finder (www.interscience.wiley.com)

Search Strategy

A search string is constructed using
*Technique, Aspect*and *mining*as main keywords, andincluded synonyms and related terms. The search string is then constructed using Boolean "**AND**" to connect the three keywords and Boolean "**OR**" to allow synonyms and word level alternatives of each keyword. The resulting search string is depicted in Listing 1.
Listing 1. Search string

Listing 1. Search string

("Technique" OR "Approach" OR "Method" OR "Methodology") AND ("Aspect mining" OR "Cross-cutting concern mining" OR "Crosscutting concern mining" OR "Code mining" OR "Separation of Concern mining" OR "Concern mining") AND NOT ("Web Mining" OR "Text Mining" OR "Early Aspect Mining")

The search string was executed in the digital libraries/indexing services to titles, abstracts and metadata presumingthat these provide a clearoutline of the work. The search scope is open as no range of study dates has been defined in the search strategy.

Study Selection Strategy

To ensure that all papers included in the review are clearly relevant to the research questions, inclusion and exclusion criteria have been defined in table 1.

Table 1. Inclusion and exclusion criteria for study selection

Inclusion criteria	Exclusion criteria
I1.Papers that deals with mining of concerns in the software in particular. I2. Papers having the objective of the study as to present a solution to the concern mining. I3. Published papers which are not short papers. I4. Peer reviewed papers.	E1. Papers which are not written in English. E2. A study that is related to challenges and issues in the migration of legacy systems to Aspect oriented systems. E3. Papers having the objective of the study as to not present a solution to the concern mining.

The study selection not only disapproves studies which are not binding the objective of the present paper, but also guarantees the quality of the study and the scope of the research e.g. inclusion criterion I3, I4 and exclusion criterion E3 ensure that the primary study data meet the standards of peer-reviewed papers. Inclusion criteria I1, I2 and exclusion criteria E1 and E2 scope the research in accordance with the research objective/motivation.

Data Extraction

The main objective of this step is to design data extraction forms with the purpose to document the data extracted from the primary studies [73]. To extract the papers which are relevant to the present study, usual details had been noted such as- (i) title, (ii) authors, (iii) abstract, (iv) technique applied (v) publication year, (vi) publication name and (vii) publication type i.e., journal/conference/workshop/book chapter. Further, the following steps are performed to ensure the level of extracted papers:

i. Use the title to remove any papers clearly not related to the research questions;
ii. Use the abstract and keywords to reject papers not related to the research questions; and
iii. Read the rest part of the papers and remove any which do not fulfill the criterion stated in Table1

At last a group of reviewers consisting of final year MCA students is formed, which repeated the three steps as enumerated above, to cross check that the result (the extracted papers) are same. The finally extracted papers of relevant studies are referred to as primary studies.

Data Synthesis

Collation and summarization of the results of selected/included primary studies is called data

synthesis [73]. The primary studies were evaluated against the evaluation framework presented in subsection 2.1.3.4 and various findings are reported in Section 2.3.

Conducting the Review

In this phase, firstly primary studies are identified in the digital libraries. The digital library Scopus has returned more primary studies than the others (262). IEEE, ACM and Springer have returned 215, 202 and 127 respectively. Finally, 806 primary studies have been gotten. Next, the selection of primary studies is done by means of reading the titles and abstracts and the application of the inclusion and exclusion criteria. As a result, a total of 126 primary studies have been done that were read completely, so the outcomes obtained were 48 studies. Among these 48 primary studies, 22 mining techniques for crosscutting concern are obtained.

Reporting the Review

With the objective of answering the RQ1, 22 aspect mining techniques were identified which mine aspects from the legacy systems. Following aspect mining techniques which mine aspects from the legacy systems are identified from the 48 primary studies, which are classified into two categories: Explorative Search Techniques and Generative Techniques.

Explorative Search Techniques

i. Lexical Pattern Matching (4)
   Aspect Browser
   Aspect Mining Tool
   MultiVisualizerAMTex
   Prism
ii. Query Language (1)
   JQUERY
iii. Concern Graph (1)
   FEAT

Generative Techniques

i. Analyzing Recurring Patterns of Execution Traces (3)
ii. Formal Concept Analysis (3)
Execution Traces (1)
Identifiers (1)
Software Repository (1)
iii. Natural Language Processing on Source Code (1)
iv. Unique Methods Detection (1)
v. Clustering of Related Methods (10)
Hierarchical Clustering of Similar Method Names (1)
Clustering Based on Method Invocation (1)
Clustering Based on Vector Space Model (5)
Clustering Based on Graph(1)
Clustering Based on Fan-In Analysis (CBFA) (1)
vi. Fan-in Analysis (1)
vii. Clone Detection (7)
Detecting Aspects Using Program Dependency Graphs and Abstract Syntax Tree (1)
Using Token Matching (2)
Using Metrics (2)
Based on AST Comparison (1)
Using DIFF Comparison Algorithm (1)
viii. Random Walk Model (2)
Experience Mining (1)
Concept Mining (1)
ix. Information Retrieval Technique (2)
Community Detection Technique (1)
Link Analysis Technique (1)
x. Association Rules (1)
xi. Method Call Tree (1)
xii. Software Repository Mining (3)
Frequent Itemset Mining (1)
Historical Code Changes (1)
Analyzing Line Co-Change (1)
xiii. Analyzing Type Hierarchies (1)
xiv. Analyzing Code Patterns (1)
xv. Similar Interaction Patterns (1)
xvi. Dataflow Abstraction (1)
xvii. Model-Driven Plan Based Approach (1)
xviii. Meta-Model (1)
xix. Idiom-Driven Approach (1)

Note: Here, the early aspects are not considered in the systematic review since the requirement documents are not available or architectures have become outdated.

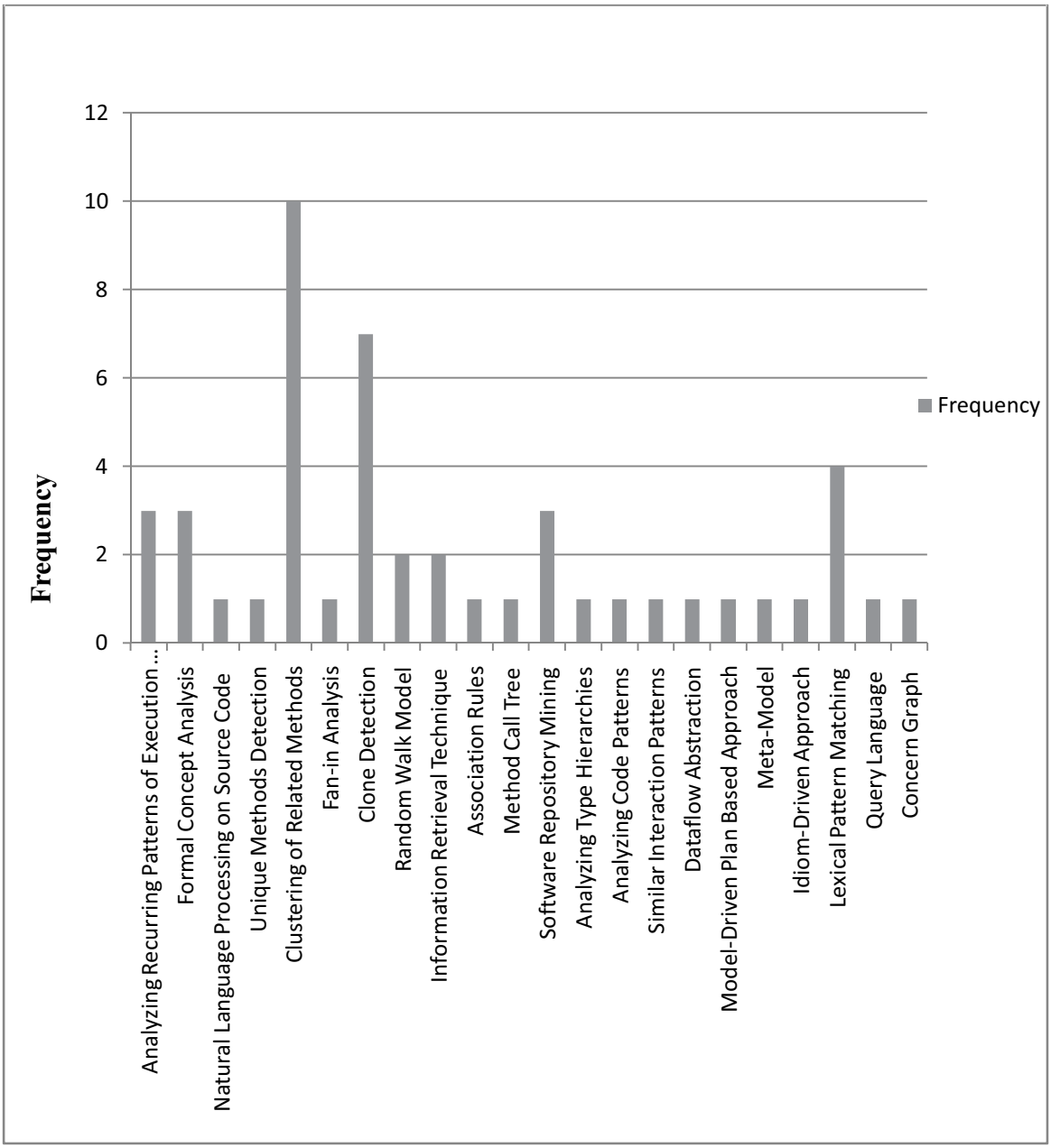

Figure 1: shows the frequency distribution of aspect mining approaches used in the migration of legacy systems to aspect oriented systems.

From the figure 1 the answer to the research question ***RQ1*** is obvious. Clustering of related methods is used maximum number of times to mining crosscutting concerns from legacy systems.

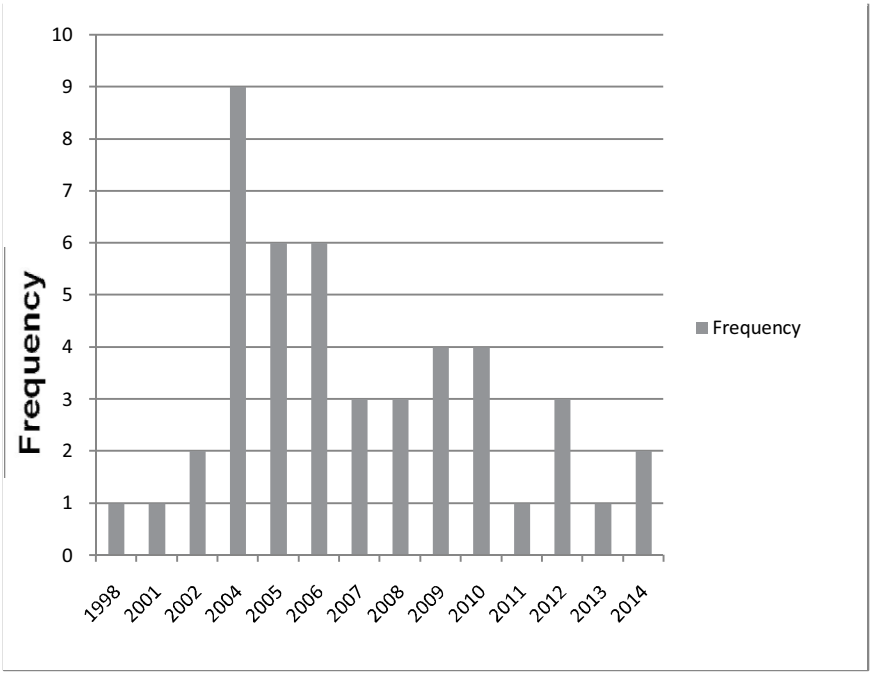

Fig 2: Yearly distribution of research articles published on aspect mining in legacy software during the period 1998-2014.

The answer to the research question ***RQ2***is obvious from the Fig 2 which clearly shows that the number of publications in aspect mining in legacy software systemsrise in the year 2004to reached its maximum level followed by the year 2005 and 2006.

Conclusion

The main contribution of this paper is to identify which approach is mostly used in the migration from legacy systems to aspect-oriented software system. This may help the researchers to provide new approaches in crosscutting concern mining area.

There are many tools which are used in combination with the crosscutting concern mining approaches. Further studies may be done to identify which technique is used to identify scattered or tangled concerns or are applied statically or dynamically.

REFERENCES

Adams, B., Jiang, Z. M., & Hassan, A. E. (2010, May). Identifying crosscutting concerns using historical code changes. In Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1(pp. 305-314). ACM.

Ajila, S. A., Gakhar, A. S., & Lung, C. H. (2013). Aspectualization of code clones—an algorithmic approach. Information Systems Frontiers, 1-17.

Baxter, I. D., Yahin, A., Moura, L., Sant'Anna, M., & Bier, L. (1998, November). Clone detection using abstract syntax trees. In Proceedings of the International Conference on Software Maintenance, (pp. 368-377). IEEE.

Bernardi, M. L., & Di Lucca, G. A. (2009, August). A role-based crosscutting concerns mining approach to evolve java systems towards AOP. In Proceedings of the joint international and annual ERCIM workshops on Principles of software evolution (IWPSE) and software evolution (Evol) workshops (pp. 63-72). ACM.

Bernardi, M. L., & Di Lucca, G. A. (2009, October). ConAn: A tool for the identification of crosscutting concerns in object oriented systems based on type hierarchy analysis. In Reverse Engineering, 2009. WCRE'09. 16th Working Conference on (pp. 319-320). IEEE.

Bisbal, J., Lawless, D., Wu, B., Grimson, J., Wade, V., Richardson, R., & O'Sullivan, D. (1997, December). An overview of legacy information system migration. In Software Engineering Conference, 1997. Asia Pacific and International Computer Science Conference 1997. APSEC'97 and ICSC'97. Proceedings (pp. 529-530). IEEE.

Breu, S. (2004, November). Towards hybrid aspect mining: Static extensions to dynamic aspect mining. In 1st Workshop on Aspect Reverse Engineering.

Breu, S., &Krinke, J. (2004, September). Aspect mining using event traces. In Automated Software Engineering, 2004. Proceedings. 19th International Conference on (pp. 310-315). IEEE.

Breu, S., & Zimmermann, T. (2006, September). Mining aspects from version history. In Automated Software Engineering, 2006. ASE'06. 21st IEEE/ACM International Conference on (pp. 221-230). IEEE.

Breu, S., Zimmermann, T., &Lindig, C. (2006, May). Mining eclipse for crosscutting concerns. In Proceedings of the 2006 international workshop on Mining software repositories (pp. 94-97). ACM.

Bruntink, M. (2004, November). Aspect mining using clone class metrics. In 1st Workshop on Aspect Reverse Engineering.

Bruntink, M., Van Deursen, A., Van Engelen, R., &Tourwe, T. (2005). On the use of clone detection for identifying crosscutting concern code. Software Engineering, IEEE Transactions on, 31(10), 804-818.

Canfora, G., Cerulo, L., & Di Penta, M. (2006, September). On the use of line co-change for identifying crosscutting concern code. In Software Maintenance, 2006. ICSM'06. 22nd IEEE International Conference on (pp. 213-222). IEEE.

Comella-Dorda, S., Wallnau, K., Seacord, R. C., & Robert, J. (2000). A survey of legacy system modernization approaches (No. CMU/SEI-2000-TN-003). Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Institute.

Dongjin, Y. U., Xiang, S. U., &Yunlei, M. U. (2014). Towards the Identification of Crosscutting Concerns: A Comprehensive Dynamic Approach Based on Execution Relations. IEICE TRANSACTIONS on Information and Systems, 97(5), 1235-1243.

El-Aziz, R. M. A., Aboutabl, A. E., &Mostafa, M. S. (2012). Clone Detection Using DIFF Algorithm for Aspect Mining. International Journal of Advanced Computer Science & Applications, 3(8).

Ganter, B., Wille, R., &Wille, R. (1999). Formal concept analysis (Vol. 284). Berlin: Springer.

Griswold, W. G., Kato, Y., & Yuan, J. J. (2000). Aspectbrowser: Tool support for managing dispersed aspects. Department of Computer Science and Engineering, University of California, San Diego.

Gybels, K., &Kellens, A. (2004, September). An experiment in using inductive logic programming to uncover pointcuts. In First European Interactive Workshop on Aspects in Software (Vol. 24).

Gybels, K., &Kellens, A. (2005, January). Experiences with identifying aspects in smalltalkusing'unique methods'. In Workshop on Linking Aspect Technology and Evolution.

Hannemann, J., &Kiczales, G. (2001, May). Overcoming the prevalent decomposition in legacy code. In Workshop on Advanced Separation of Concerns (Vol. 167).

He, L., Bai, H., Zhang, J., & Hu, C. (2005). Amuca algorithm for aspect mining.Software Engineering and Knowledge Engineering (SEKE).

Huang, J., Lu, Y., & Yang, J. (2010, August). Aspect mining using link analysis. In Frontier of Computer Science and Technology (FCST), 2010 Fifth International Conference on (pp. 312-317). IEEE.

Hunt, J. W., &McIlroy, M. D. (1976). An algorithm for differential file comparison. Bell Laboratories.

Ishio, T., Miyake, T., & Inoue, K. (2008, October). Mining coding patterns to detect crosscutting concerns in java programs. In Reverse Engineering, 2008. WCRE'08. 15th Working Conference on (pp. 123-132). IEEE.

Iyer, V. N. (2008). Legacy Modernization- Modernize and Scale. View point, Infosys.

Janzen, D., & De Volder, K. (2003, March). Navigating and querying code without getting lost. In Proceedings of the 2nd international conference on Aspect-oriented software development (pp. 178-187). ACM.

Kamiya, T., Kusumoto, S., & Inoue, K. (2002). CCFinder: a multilinguistic token-based code clone detection system for large scale source code. Software Engineering, IEEE Transactions on, 28(7), 654-670.

Kellens, A., Mens, K., &Tonella, P. (2007). A survey of automated code-level aspect mining techniques. In Transactions on aspect-oriented software development IV (pp. 143-162). Springer Berlin Heidelberg.

Khadka, R., Batlajery, B. V., Saeidi, A., Jansen, S., &Hage, J. (2014, May). How do professionals perceive legacy systems and software modernization?. In ICSE (pp. 36-47).

Komondoor, R., &Horwitz, S. (2001). Using slicing to identify duplication in source code. In Static Analysis (pp. 40-56). Springer Berlin Heidelberg.

Krinke, J., &Breu, S. (2004, November). Control-flow-graph-based aspect mining. In 1st Workshop on Aspect Reverse Engineering.

Maisikeli, S. G., & Mitropoulos, F. J. (2010, October). Aspect mining using self-organizing maps with method level dynamic software metrics as input vectors. In Software Technology and Engineering (ICSTE), 2010 2nd International Conference on (Vol. 1, pp. V1-212). IEEE.

Malinova, A. (2010). Approaches and techniques for legacy software modernization. Plovdiv University "PaissiiHilendarski", Bulgaria Scientific Works, 37(3), Mathematics.

Marco D'Ambros and Michele Lanza. Reverse engineering with logical coupling. In Proceedings of the Working Conference on Reverse Engineering (WCRE), pages 189–198. IEEE, 2006.

Marin, M., Deursen, A. V., &Moonen, L. (2007). Identifying crosscutting concerns using fan-in analysis. ACM Transactions on Software Engineering and Methodology (TOSEM), 17(1), 3.

Marin, M., Van Deursen, A., &Moonen, L. (2004, November). Identifying aspects using fan-in analysis. In Reverse Engineering, 2004. Proceedings. 11th Working Conference on (pp. 132-141). IEEE.

Marin, M., van Deursen, A., Moonen, L., & van der Rijst, R. (2009). An integrated crosscutting concern migration strategy and its semi-automated application to JHotDraw. Automated Software Engineering, 16(2), 323-356.

Mayrand, J., Leblanc, C., & Merlo, E. M. (1996, November). Experiment on the automatic detection of function clones in a software system using metrics. In Proceedings of the Software Maintenance, International Conference on (pp. 244-253). IEEE.

Mcfadden, R. R., & Mitropoulos, F. J. (2012, March). Aspect mining using model-based clustering. InProceedings of the Southeast Conference of IEEE (pp. 1-8). IEEE.

Mens, K., Kellens, A., &Krinke, J. (2008, October). Pitfalls in aspect mining. In Reverse Engineering, 2008. WCRE'08. 15th Working Conference on (pp. 113-122). IEEE.

Moldovan, G. S., &Serban, G. (2006, March). Aspect mining using a vector-space model based clustering approach. In Proceedings of Linking Aspect Technology and Evolution (LATE) Workshop (pp. 36-40).

Mulder, F., &Zaidman, A. (2010, September). Identifying crosscutting concerns using software repository mining. In Proceedings of the Joint ERCIM Workshop on Software Evolution (EVOL) and International Workshop on Principles of Software Evolution (IWPSE) (pp. 23-32). ACM.

Nguyen, T. T., Nguyen, H. V., Nguyen, H. A., & Nguyen, T. N. (2011, May). Aspect recommendation for evolving software. In Proceedings of the 33rd International Conference on Software Engineering (pp. 361-370). ACM.

Nora, B., & Ghoul, S. (2006). A model-driven approach to aspect mining. In 27th International Conference on Software Engineering (pp. 361-370).

Prechelt, L., Malpohl, G., &Philippsen, M. (2002). Finding plagiarisms among a set of programs with JPlag. J. UCS, 8(11), 1016.

"Project Bauhaus," http://www.bauhaus-stuttgart.de, 2005.

Qu, L., & Liu, D. (2007, April). Aspect mining using method call tree. In Multimedia and Ubiquitous Engineering, 2007. MUE'07. International Conference on (pp. 407-412). IEEE.

Robillard, M. P., & Murphy, G. C. (2007). Representing concerns in source code. ACM Transactions on Software Engineering and Methodology (TOSEM),16(1), 3.

Serban, G., &Cojocar, G. S. (2007). A New Graph-Based Approach in Aspect Mining. In Proceedings of the International Conference on Knowledge Engineering (pp. 252-260).

Serban, G., & Moldovan, G. S. (2006). A Graph Algorithm for Identification of Crosscutting Concerns. StudiaUniversitatis Babes-Bolyai, Informatica, LI (2), 53-60.

Serban, G., & Moldovan, G. S. (2006, October). A new genetic clustering based approach in aspect mining. In Proceedings of the 8th WSEAS international conference on Mathematical methods and computational techniques in electrical engineering (pp. 135-140).

Serban, G., & Moldovan, G. S. (2006, September). A new k-means based clustering algorithm in aspect mining. In Eighth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, 2006. SYNASC'06. (pp. 69-74). IEEE.

Shepherd, D., & Pollock, L. (2005, March). Interfaces,

aspects, and views. In Linking Aspect Technology and Evolution (LATE) Workshop (pp. 1-6).

Shepherd, D., Gibson, E., & Pollock, L. L. (2004, April). Design and Evaluation of an Automated Aspect Mining Tool. In Software Engineering Research and Practice (pp. 601-607).

Shepherd, D., Pollock, L., &Tourwé, T. (2005, May). Using language clues to discover crosscutting concerns. In ACM SIGSOFT Software Engineering Notes(Vol. 30, No. 4, pp. 1-6). ACM.

Tarr, P., Ossher, H., Harrison, W., & Sutton Jr, S. M. (1999, May). N degrees of separation: multi-dimensional separation of concerns. In Proceedings of the 21st international conference on Software engineering (pp. 107-119). ACM.

Tekinerdogan, B., &Aksit, M. (1998). Deriving design aspects from conceptual models. Lecture Notes in Computer Science, 1543, 410-413.

Tonella, P., &Ceccato, M. (2004, November). Aspect mining through the formal concept analysis of execution traces. In 11th Working Conference on Reverse Engineering, 2004. Proceedings. (pp. 112-121). IEEE.

Tourwe, T., &Mens, K. (2004, September). Mining aspectual views using formal concept analysis. In Fourth IEEE International Workshop on Source Code Analysis and Manipulation, 2004. (pp. 97-106). IEEE.

Trifu, M. (2008, April). Using dataflow information for concern identification in object-oriented software systems. In Software Maintenance and Reengineering, 2008. CSMR 2008. 12th European Conference on (pp. 193-202). IEEE.

Vidal, S., Abait, E. S., Marcos, C., Casas, S., &Díaz Pace, J. A. (2009, March). Aspect mining meets rule-based refactoring. In Proceedings of the 1st workshop on Linking aspect technology and evolution (pp. 23-27). ACM.

Yuen, I., &Robillard, M. P. (2007, March). Bridging the gap between aspect mining and refactoring. In Proceedings of the 3rd workshop on Linking aspect technology and evolution (p. 1). ACM.

Zhang, C., & Jacobsen, H. A. (2004, October). Prism is research in aspect mining. In Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications (pp. 20-21). ACM.

Zhang, C., & Jacobsen, H. A. (2007, March). Efficiently mining crosscutting concerns through random walks. In Proceedings of the 6th international conference on Aspect-oriented software development (pp. 226-238). ACM.

Zhang, C., & Jacobsen, H. A. (2012). Mining crosscutting concerns through random walks. Software Engineering, IEEE Transactions on, 38(5), 1123-1137.

Zhang, C., Gao, D., & Jacobsen, H. A. (2002). Extended aspect mining tool.URL: http://www. eecg. utoronto. ca/ czhang/amtex.

Zhang, D., Guo, Y., & Chen, X. (2008, September). Automated aspect recommendation through clustering-based fan-in analysis. In 23rd IEEE/ACM International Conference on Automated Software Engineering, 2008. (pp. 278-287). IEEE.

Zhu, J., Huang, J., Zhou, D., Carminati, F., Zhang, G., & He, Q. (2013). Identifying composite crosscutting concerns through semi supervised learning. Software: Practice and Experience, vol. 44, pp (1525–1545).

Kitchenham, B>A>charters,s.: guidelines(2007

J. Biolchini, P.G. Mian, A.C. Natatli, G.H. Travassos, Systematic Review in Software Engineering: Relevance and Utility, PESC-COPPE/UFRJ, Brazil, 2005, <http://cronos.cos.ufrj.br/publicacoes/reltec/es67905.pdf>.

T. Dyba, Experiences of Undertaking Systematic Reviews, SINTEF ICT, Queensland, 2005.

B. Kitchenham, Procedures for Performing Systematic Reviews, Technical Report TR/SE-0401, Department of Computer Science, Keele University and National ICT, Australia, Ltd., 2004, http://www.elsevier.com/framework_products/promis_misc/inf-systrev.pdf