

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

# INFORMATION TECHNOLOGY JOURNAL

**ANSI***net*

Asian Network for Scientific Information  
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

## A Review on Aspect Oriented Implementation of Software Product Lines Components

Fazal-e-Amin, Ahmad Kamil Mahmood and Alan Oxley  
Department of Computer and Information Sciences,  
Universiti Teknologi PETRONAS, 31750, Tronoh Perak, Malaysia

---

**Abstract:** Aspect orientation is an emerging technique which provides a solution to the implementation of concerns that span more than one module, thus enhancing modularity. Modularity is directly related to reusability. The whole point of having a software product line is so that software assets can be reused to develop multiple individual products. This co-relation between aspect-oriented techniques and software product lines motivates the development of aspect-oriented product line components. Several researchers have presented their work on the development of core assets using aspect-oriented techniques. These studies used aspect-oriented techniques for an improved separation of a concern, the treatment of crosscutting concerns, the inclusion of optional/alternative features and for variability management. In this study, an attempt is made to review the research in this field, highlighting the application domains along with other pertinent results. Furthermore, the study also tries to highlight the opportunities and challenges for carrying out the aspect-oriented implementation of software product line components.

**Key words:** Aspect-oriented, software product line components, software engineering

---

### INTRODUCTION

Aspect-oriented techniques provide a solution to the crosscutting concerns problem. A system can have different concerns; a concern is an area of interest or property of a system that must be implemented in order for there to be a successful solution to a problem (Elrad *et al.*, 2001). Traditional software engineering is involved in the identification of concerns and these concerns are used to modularize a system.

The concept of encapsulation in software engineering leads us to attempt to implement each concern as a separate module. This phenomenon is also known as the separation of concern. It is not always possible to map each concern to a separate module.

Kiczales (1996) described some of the points relating to concerns. The implementation of a single concern over more than one module is termed crosscutting, it creates the problem of concern/code tangling and scattering. Concern tangling is the situation when more than one concern is implemented in a single module and concern scattering is when one concern is implemented in multiple modules. Crosscutting concerns could be, for example, the system wide quality requirements. Kiczales divides concerns into two categories: Aspects and Components. If a concern can be cleanly encapsulated in a module it will be a component and it will be an aspect if the concern crosscuts and cannot be cleanly implemented in a single

module; these separate specifications of aspects and components are then combined to provide the solution by the process of weaving. Aspect-Oriented Software Engineering introduced a new mechanism to modularize a system and separate the crosscutting concerns.

Software Product Lines (SPLs) are getting more attention from the software community due to the potential benefits, such as improved productivity, increased quality, decreased effort, reduced cost, faster time to market and an increased ability to move into new markets (Linden *et al.*, 2007; Clements and Northrop, 2001). The SPLs are based on a product line infrastructure, central to which is a collection of reusable core assets.

Clements and Northrop (2001) made some points relevant to SPLs. Unlike traditional ways of reusing software, SPLs are intended to reuse all the assets developed during the software development life cycle. An SPL targets a specific domain and captures commonalities between individual products and manages variability between them. Variability represents the distinguishing characteristics of the individual products. The success of SPLs depends on efficient variability management. The SPLs promote the systematic reuse of core assets. Unlike traditional ways of reuse in software development, every artefact developed during the software development process is intended to be reused by more than one product. These reusable software artefacts are referred to as core assets. Core assets include almost all the artefacts

developed during the software development process, such as requirement documents, architecture, use cases and components.

Various individual products are derived from the core assets. These products are distinguished from each other on the basis of their features. Members of a product line have some common and distinctive features. As we stated above, commonality is the set of common features shared by all products in a product line and variability represents the features specific to individual products.

The SPLs focus on the systematic reuse of core assets. The underlying tenet is development for the reuse and development with reuse (Linden *et al.*, 2007). Software reusability is the key feature of SPLs.

The authors' previous work includes a limited survey of aspect oriented implementations and also proposes a framework for assessing the reusability of components used in such implementations (Fazal-e-Amin and Mahmood, 2009). This framework makes use of a reusability attribute model, the details of which have been expanded by Fazal-e-Amin and Mahmood (2009).

There exists some reviews in the field of SPLs, such as (Chen *et al.*, 2009), which presents a systematic review of variability management in software products. There also exists some reviews in the field of aspect-oriented modelling approaches, such as (Chitchyan *et al.*, 2005). We decided to conduct a limited systematic search in order to find any existing reviews about the aspect-oriented implementation of SPL components. We searched Scopus, Google scholar, IEEE Xplore and the ACM portal digital library but were unable to find a review on this topic.

## MATERIALS AND METHODS

A systematic literature review process is presented in (Brereton *et al.*, 2007) with the lesson learned by considering several research reviews in software engineering. We have followed this process to conduct the review. An adaptation of this process is shown in Fig. 1.

The review process has three main activities and ten sub activities. The research questions answered are:

- Question 1:** What domains have benefited from the use of aspect-oriented techniques?
- Question 2:** What motivates us to use aspect orientation as the product line implementation technology?
- Question 3:** What results have been achieved by the use of aspect-oriented technology for SPL component development?

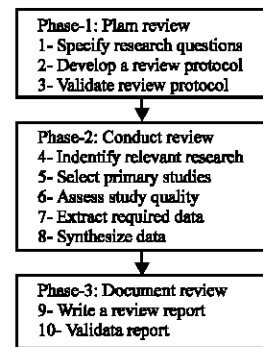


Fig. 1: Adapted from Brereton *et al.* (2007)

**Question 4:** What are the opportunities and challenges for carrying out the aspect-oriented implementation of SPL components?

In line with the second step of phase-1, a protocol is developed for this review. The protocol includes the sources to be used for this review, the period of time and the criteria to search and select the research studies. Furthermore, the protocol is reviewed and some changes were made and then validated by the researchers.

The sources selected to be used for this review are Scopus, Google scholar, IEEE Xplore and the ACM portal digital library. Additionally we have looked at the ICSE, AOSD, AOPLE, SPLC conferences and workshops. The search is performed to find publications during the years 2000 to 2009. Table 1 show the year wise results of the search process.

Research studies are collected by using different queries on the sources to find publications which discuss the use of aspect orientation at the implementation level of SPL development. The collection of research study is refined on the basis of keywords and abstracts under the protocol defined and validated earlier. The quality criteria set to assess the studies was to include a research study only if it contained a concrete example to demonstrate the use of aspect orientation in product line development with results. The required data was then extracted from the research study and a synthesis was performed on data. Finally, the review report is written and validated.

## RESULTS

The results of the review are presented in this section. The results are categorized according to the questions posed earlier.

**What domains have benefited from the use of aspect-oriented techniques?:** The results of the review (Fig. 2) reveal the following product line domains in which aspect

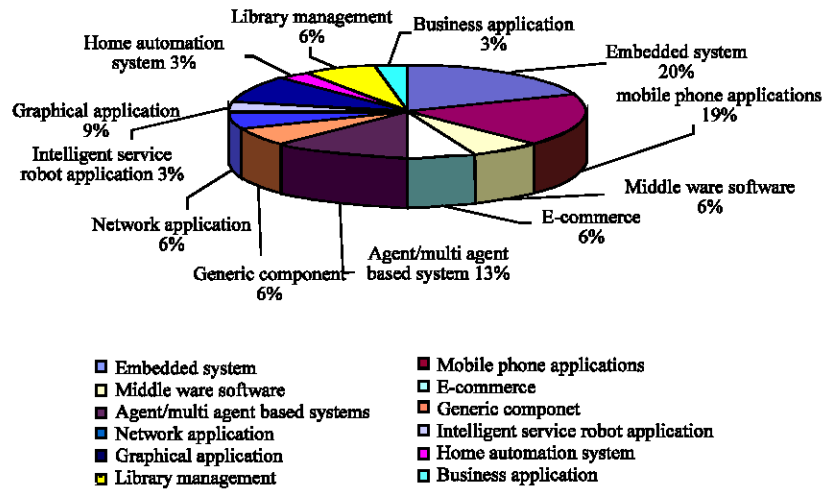


Fig. 2: Domain wise representation of results

Table 1: Year wise search results

Year	No. of publication
2000	2
2001	0
2002	0
2003	2
2004	6
2005	6
2006	8
2007	4
2008	7
	20091
Total	36

orientation is used as an implementation technology. The most prominent use of aspect orientation is in the domain of embedded systems (20%). For instance, an implementation of a weather station product line using Aspect C++ is presented by Beuche and Spinczyk (2003a, b). Other applications in this domain include an operating system component (Lohmann *et al.*, 2005) and an implementation of an application for microwave oven software (Seung-Hyun and Choi, 2006).

The mobile phone application domain (19%) includes the implementation of mobile phone games (Alves *et al.* 2005, 2006; Alves, 2005; Kulesza *et al.*, 2006; Figueiredo *et al.*, 2008), the optional auto complete text feature T9 (Anastasopoulos and Muthig, 2004) and a multimedia application (Figueiredo *et al.*, 2008).

Thirteen percent of the selected publications implemented agent/ multi-agent oriented systems. An agent oriented system is described by Griss (2000b). Multi-agent oriented product lines are described by Kulesza *et al.* (2006), Nunes *et al.* (2009) and Sant'Anna *et al.* (2008). The latter study also gives a modularity assessment of aspect-oriented product lines.

The development of graphical applications (9%) is discussed by Santos *et al.* (2006, 2007) and Lopez-Herrejon and Batory (2006).

Library management systems (6%) are implemented by Peng *et al.* (2008) and Zhang *et al.* (2008a). Zhang and Jacobsen (2004) and Colyer and Clement (2004) middleware software applications(6%) are demonstrated. Implementation of a business application (3%) is discussed by Pohl *et al.* (2008) and e-commerce applications (6%) are developed by Griss (2000a) and Nyßen *et al.* (2005).

Generic components (6%) implementation is stated by Muthig and Patzke (2003) and (Loughran and Rashid, 2004).

Network application (6%) is developed by Apel and Batory (2006) and Apel *et al.* (2008). An intelligent service robot application (3%) is shown by Lee *et al.* (2006).

In Voelter and Groher (2007) implementation of a home automation system (3%) is discussed.

What motivates us to use aspect orientation as the product line implementation technology?: The motivations for using aspect orientation as the product line technology (Fig. 3), observed from the selected study, are presented in this section. A majority of the selected publications used aspect orientation to manage variability (24%) such as Beuche and Spinczyk (2003b), Muthig and Patzke (2003), Anastasopoulos and Muthig (2004), Mezini and Ostermann (2004) Alves (2005), Nyßen *et al.* (2005), Voelter and Groher (2007) and Zhang *et al.* (2008a, b).

The aspect-oriented technique is used for the removal of crosscutting concerns (14%) (Mezini and Ostermann, 2004; Kulesza *et al.*, 2006; Zhang *et al.*, 2008a, b; Voelter and Groher, 2007).

Another prominent usage is for inclusion of optional/alternative features (14%) that is presented by

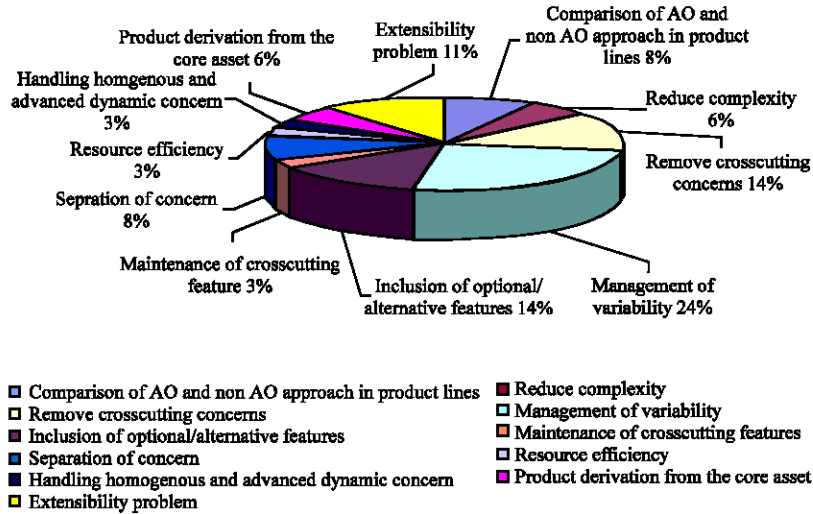


Fig. 3: Motivation to use AO in SPL

Kulesza *et al.* (2006), Peng *et al.* (2008), Figueiredo *et al.* (2008), Sant’Anna *et al.* (2008) and Nunes *et al.* (2009).

The extensibility problem (11%) is solved by Kulesza *et al.* (2006), Santos *et al.* (2006), Pohl *et al.* (2008) and Zhang *et al.* (2008b).

Other motivations include a comparison of AO and non AO approaches (8%), separation of concerns (8%), complexity reduction (6%), product derivation from the core asset (6%), maintenance of crosscutting features (3%), handling of homogenous and advanced dynamic concerns (3%) and resource efficiency (3%). The results are shown in Table 2.

**What results have been achieved by the use of aspect-oriented technology for SPL component development?:**

The review of the selected publications shows the following advantages (Fig. 4) of using aspect orientation as the product line implementation technique.

Most of the selected publications report about variability management (32%). These include Beuche and Spinczyk (2003b), Muthig and Patzke (2003), Anastasopoulos and Muthig (2004), NyBen *et al.* (2005) and Voelter and Groher (2007).

The advantage of enhanced modularity (21%) is identified by Mezini and Ostermann (2004), Santos *et al.* (2006), Voelter and Groher (2007), Figueiredo *et al.* (2008), Pohl *et al.* (2008) and Nunes *et al.* (2009).

Removal of crosscutting (18%) is stated as an advantage by Mezini and Ostermann (2004), Kulesza *et al.* (2006), Zhang *et al.* (2008a, b) and Voelter and Groher (2007).

Some of the publications claimed improved configurability (11%). Other advantages include reduced complexity (7%), efficient product derivation from core assets (7%) and increased changeability (4%).

Table 2: Results of question 1 and 2

Question	Categories	No. of publications
1: What domains have benefited from the use of aspect-oriented techniques?	Embedded system	6
	Extensibility problem	4
	Mobile phone application	6
	Middleware software	2
	E-commerce	2
	Agent/multi agent based system	4
	Generic component	2
	Network application	2
	intelligent service	1
	Robot application	1
	Graphical application	3
	Home automation system	1
	Library management	2
	Business application	1
Total	36	
2: What motivates us to use aspect orientation as the product line implementation technology?	Comparison of AO and non AO approaches in product lines	3
	Reduce complexity	2
	Remove crosscutting concerns	5
	Management of variability	9
	Inclusion of optional/alternative features	5
	Maintenance of crosscutting features	1
	Separation of concern	3
	Resource efficiency	1
	Handling homogeneous and advanced dynamic concern	1
	Product derivation from the core asset	2
	Extensibility Problem	4
	Total	36

**What are the opportunities and challenges for carrying out the aspect-oriented implementation of SPL components?:**

The following challenges and opportunities are identified from the review.

**Challenges:** The most significant challenge to the aspect-oriented implementation of product lines is

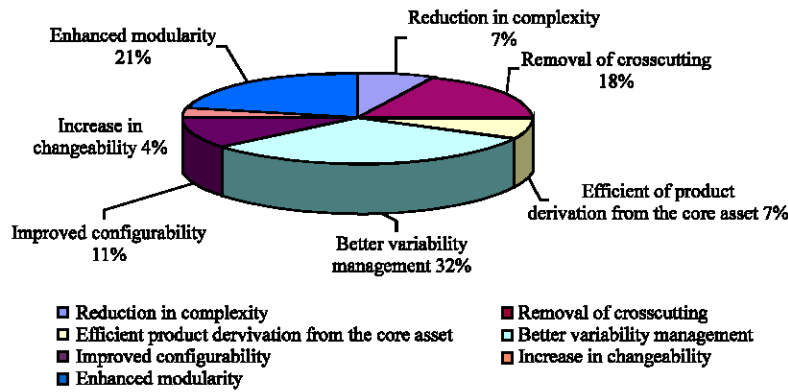


Fig. 4: Results discussed in selected study

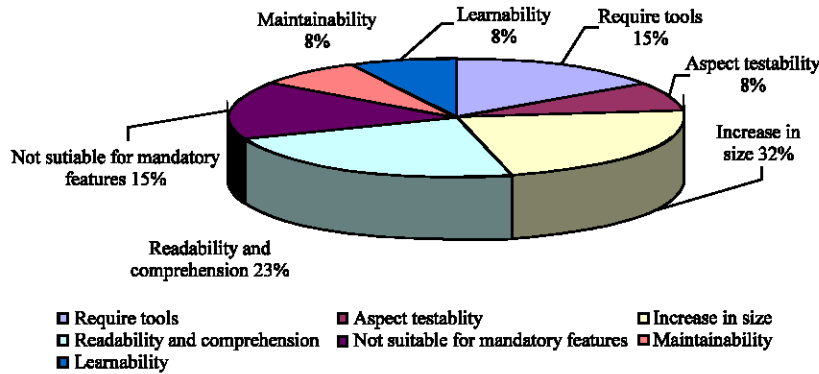


Fig. 5: Challenges mentioned in selected study

increase in size (23%), which is reported by Alves *et al.* (2006), Sant’Anna *et al.* (2008) and Nunes *et al.* (2009). This increase in size restricts the use of aspect orientation in devices run on constrained resources. The other major challenge is readability and comprehension (23%) stated by Nyßen *et al.* (2005), Kastner *et al.* (2007) and Sant’Anna *et al.* (2008).

It is highlighted by Figueiredo *et al.* (2008) that aspect orientation is not suitable for implementing mandatory features (15%) nor to changing mandatory features to optional/alternative ones (Fig. 5).

The requirement of tools (15%) or unavailability of tools is mentioned Muthig and Patzke (2003) and Zhang and Jacobsen (2004).

The other challenges include learn-ability (8%), aspect testability (8%) and maintainability of crosscutting features (8%). The results of question 3 and 4 are shown in Table 3.

**Opportunities:** The results mentioned by Anastasopoulos and Muthig (2004), Santos *et al.* (2006) and Pohl *et al.* (2008) suggested that the opportunity to reuse increases by using aspect orientation as the product line implementation technology.

Table 3: Results of question 3 and 4

Question	Categories	No. of publications
3: What results have been achieved by the use of aspect-oriented technology for SPL component development?	Reduction in complexity	2
	Removal of crosscutting	5
	Efficient product derivation from the core asset	2
	Better variability management	9
	Improved Configurability	3
	Increase in Changeability	1
	Enhanced Modularity	6
	Total	28*
4: What are the challenges for carrying out the aspect-oriented implementation of SPL components?	Require tools	2
	Aspect testability	1
	Increase size	3
	Readability and comprehension	3
	Not suitable for mandatory feature	1
	Maintainability	1
	Learnability	1
	Total	12*

\*Some of the selected publications contain no information to answer these questions

## CONCLUSIONS AND FUTURE WORK

Recent study proposes the use of aspect-oriented technology for SPL implementation. In this study we have reviewed these works to have an insight into the

motivations to use aspect orientation as the product line implementation technology. The product line domains which have benefited from aspect orientation are categorized. The results, opportunities and challenges highlighted in selected publications are reported in this review.

The results of the review show that most of the product lines that have been implemented using aspect orientation are embedded systems (20%) and mobile phone software (19%), on the other hand some studies (23%) indicate that aspect orientation is not suitable for product lines which run on constrained systems. Therefore, it is suggested to conduct more studies to investigate the usage of aspect orientation in product line development for devices with limited resources.

The results exhibit the prominent motivations for the use of aspect orientation as product line implementation technology are variability management (24%), inclusion of optional/alternative features (14%) and removal of crosscutting features (14%).

The major benefits of using aspect orientation as the product line technology are enhanced modularity (21%) and better variability management (32%).

The results revealed that the increase in size (23%) and the readability/comprehension (23%) are the key challenges faced by the aspect-oriented product lines.

Although, our review has explored the field, further studies are needed to confirm the obtained results. Future work includes the extension of this review by including more sources. Furthermore, we are aiming to investigate the challenges of aspect-oriented product line implementations and their solutions.

## REFERENCES

- Alves, V., 2005a. Identifying variations in mobile devices. *J. Object Technol.*, 4: 47-52.
- Alves, V., P. Matos, L. Cole, P. Borba and G. Ramalho, 2005b. Extracting and evolving mobile games product lines. *Proceedings of the 9th International Software Product Line Conference*, Sept. 2005, Springer-Verlag, pp: 70-81.
- Alves, V., R. Gheyi, T. Massoni, U. Kulesza, P. Borba and C. Lucena, 2006. Refactoring product lines. *Proceedings of the 5th International Conference on Generative Programming and Component Engineering*, Oct. 22-26, Portland, Oregon, USA., pp:201-210.
- Anastasopoulos, M. and D. Muthig, 2004. An Evaluation of Aspect-Oriented Programming as a Product Line Implementation Technology. In: *Software Reuse: Methods, Techniques and Tools*, Bosch, J. and C. Krueger (Eds.). Springer-Verlag, Berlin, Heidelberg, ISBN: 978-3-540-22335-1, pp: 141-156.
- Apel, S. and D. Batory, 2006. When to use features and aspects?: A case study. *Proceedings of the 5th International Conference on Generative Programming and Component Engineering*. Oct. 22-26, ACM., Portland, Oregon, USA., pp: 59-68.
- Apel, S., T. Leich and G. Saake, 2008. Aspectual feature modules. *IEEE Trans. Softw. Eng.*, 34: 162-180.
- Beuche, D. and O. Spinczyk, 2003a. Aspect-oriented product line development in constrained environments. *Proceedings of the Workshop on Reuse in Constrained Environments OOPSLA (RICE'03), OOPSLA*, pp: 1-2.
- Beuche, D. and O. Spinczyk, 2003b. Variant management for embedded software product lines with pure::Consul and AspectC++. *Proceedings of the Companion of the 18th Annual ACM SIGPLAN Conference on Object-oriented Programming, Systems, Languages and Applications*, Oct. 26-30, Anaheim, CA, USA., pp: 108-109.
- Brereton, P., B.A. Kitchenham, D. Budgen, M. Turner and M. Khalil, 2007. Lessons from applying the systematic literature review process within the software engineering domain. *J. Syst. Softw.*, 80: 571-583.
- Chen, L., M.A. Babar and N. Ali, 2009. Variability management in software product lines: A systematic review. *Proceedings of the 13th International Software Product Lines Community (SPLC'09)*, San Francisco, CA, USA., pp: 81-90.
- Chitchyan, R., A. Rashid, P. Sawyer, A. Garcia and M.P. Alarcon *et al.*, 2005. Survey of aspect-oriented analysis and design approaches. *Proceedings of the Technical Report AOSD-Europe-ULANC-9*, May 18, AOSD-Europe, pp: 1-259.
- Clements, P. and L. Northrop, 2001. *Software Product Lines: Practices and Patterns*. Addison-Wesley Longman Publishing Co. Inc., New York.
- Colyer, A. and A. Clement, 2004. Large-scale AOSD for middleware. *Proceedings of the 3rd International Conference on Aspect-oriented Software Development*, March 22-24, Lancaster, UK., pp: 56-65.
- Elrad, T., M. Aksit, G. Kiczales, K. Lieberherr and H. Ossher, 2001. Discussing aspects of AOP. *Commun. ACM*, 44: 33-38.
- Fazal-e-Amin and A.K. Mahmood, 2009. A survey and proposed reusability assessment framework for aspect oriented product line core assets. *Proceedings of the IEEE Student Conference on Research and Development (SCOReD)*, Nov. 16-18, IEEE Explore, London, pp: 192-195.

- Figueiredo, E., N. Cacho, C. Sant'Anna, M. Monteiro and U. Kulesza *et al.*, 2008. Evolving software product lines with aspects: An empirical study on design stability. Proceedings of the 30th International Conference on Software Engineering, May 10-18, Leipzig, Germany, pp: 261-270.
- Griss, M.L., 2000a. Implementing Product-Line Features with Component Reuse. In: Software Reuse: Advances in Software Reusability, Frakes, W.B. (Eds.). LNCS., 1844, Springer-Verlag, Berlin, Heidelberg, ISBN: 978-3-540-67696-6, pp: 453-468.
- Griss, M.L., 2000b. Implementing product-line features by composing aspects. Proceedings of the 1st Conference on Software Product Lines : Experience and Research Directions: Experience and Research Directions, (SPL'00), Denver, Colorado, United States Kluwer Academic Publishers, pp: 271-288.
- Kastner, C., S. Apel and D. Batory, 2007. A case study implementing features using aspectJ. Proceedings of the 11th International Software Product Line Conference, Sept. 10-14, IEEE Computer Society, Washington, DC, USA., pp: 223-232.
- Kiczales, G., 1996. Aspect-oriented programming. ACM Comput. Surv., 28: 154-154.
- Kulesza, U., V. Alves, A. Garcia, C. de Lucena and P. Borba, 2006. Improving Extensibility of Object-Oriented Frameworks with Aspect-Oriented Programming. In: Reuse of Off-the-Shelf Components, Morisio, M. (Eds.). LNCS., 4039, Springer-Verlag, Berlin, Heidelberg, ISBN: 978-3-540-34606-7, pp: 231-245.
- Lee, K., K.C. Kang, M. Kim and S. Park, 2006. Combining feature-oriented analysis and aspect-oriented programming for product line asset development. Proceedings of the 10th International on Software Product Line Conference, Aug. 21-24, IEEE Computer Society, Washington, DC, USA., pp: 103-112.
- Linden, F. J., K. Schmid and E. Rommes, 2007. Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering. Springer-Verlag Inc., New York, ISBN: 978-3-540-71436-1, pp: 91-91.
- Lohmann, D., O. Spinczyk and W. Schröder-Preikschat, 2005. On the configuration of non-functional properties in software product lines. Proceedings of the ACP4IS Workshop, (AOSD'05), Germany, pp: 1-6.
- Lopez-Herrejon, R.E. and D. Batory, 2006. From crosscutting concerns to product lines: A function composition approach. Technical Report TR-06-24, Department of Computer Sciences, University of Texas at Austin. <http://www.cs.utexas.edu/ftp/pub/techreports/tr06-24.pdf>.
- Loughran, N. and A. Rashid, 2004. Framed Aspects: Supporting Variability and Configurability for AOP. In: Software Reuse: Methods, Techniques and Tools, Bosch, J. and C.C. Krueger (Eds.). LNCS., 3107, Springer-Verlag, Berlin, Heidelberg, ISBN: 978-3-540-22335-1, pp: 127-140.
- Mezini, M. and K. Ostermann, 2004. Variability management with feature-oriented programming and aspects. SIGSOFT Softw. Eng. Notes, 29: 127-136.
- Muthig, D. and T. Patzke, 2003. Generic Implementation of Product Line Components. In: Objects, Components, Architectures, Services and Applications for a Networked World, Aksit, M., M. Mezini and R. Unland (Eds.). LNCS. 2591, Springer-Verlag, Berlin, Heidelberg, ISBN: 978-3-540-00737-1, pp: 313-329.
- Nunes, C., U. Kulesza, C. Sant'Anna, I. Nunes and C. Lucena, 2008. On the modularity assessment of aspect-oriented multi-agent systems product lines: A quantitative study. Int. J. Agent-Oriented Softw. Eng., 2: 34-61.
- Nunes, I., C. Nunes, U. Kulesza and C. Lucena, 2009. Developing and evolving a multi-agent system product line: An exploratory study. Proceedings of the Agent-Oriented Software Engineering IX: 9th International Workshop, May 12-13, 2008, Estoril, Portugal Springer-Verlag, pp: 228-242.
- Nyßen, A., S. Tyszberowicz and T. Weiler, 2005. Are aspects useful for managing variability in software product lines? A case study. Proceedings of the Aspects and Product Lines Workshop, (SPLC'05), pp: 1-7.
- Peng, X., L. Shen and W. Zhao, 2008. Feature Implementation Modeling Based Product Derivation in Software Product Line. In: High Confidence Software Reuse in Large Systems, Mei, H. (Eds.). LNCS., 5030, Springer-Verlag, Berlin, Heidelberg, ISBN: 978-3-540-68062-8, pp: 142-153.
- Pohl, C., A. Charfi, W. Gilani, S. Gobel and B. Grammel *et al.*, 2008. Adopting aspect-oriented software development in business application engineering. Proceedings of the Aspect-Oriented Software Development, (AOSD'08), Belgium, pp: 1-10.
- Santos, A.L., A. Lopes, F. de Ciências and K. Koskimies, 2006. Modularizing framework hot-spots using aspects. Proceedings of the 11th Spanish Conference on Software Engineering and Databases, (SCSED'06), Brazil, pp: 389-399.
- Santos, A.L., A. Lopes and K. Koskimies, 2007. Framework specialization aspects. Proceedings of the 6th International Conference on Aspect-Oriented Software Development, March 12-16, Vancouver, British, Columbia, Canada, ACM, pp: 14-24.



- Sant'Anna, C., C. Lobato, U. Kulesza, A. Garcia, C. Chavez and C. Lucena, 2008. On the modularity assessment of aspect-oriented multi-agent systems product lines: A quantitative study. *Int. J. Agent-Oriented Softw. Eng.*, 2: 34-61.
- Seung-hyun, H. and E.M. Choi, 2006. Representation of variability in software product line using aspect-oriented programming. *Proceedings of the 4th International Conference on Software Engineering Research, Management and Applications*, Aug. 09-11, IEEE Computer Society, Washington, DC, USA., pp: 66-73.
- Voelter, M. and I. Groher, 2007. Product line implementation using aspect-oriented and model-driven software development. *Proceedings of the 11th International Software Product Line Conference*, Sept. 10-14, IEEE Computer Society, pp: 233-242.
- Zhang, C. and H.A. Jacobsen, 2004. Resolving feature convolution in middleware systems. *ACM SIGPLAN Not.*, 39: 188-205.
- Zhang, J., X. Cai and G. Liu, 2008a. Mapping features to architectural components in aspect-oriented software product lines. *Proc. Int. Conf. Comput. Sci. Softw. Eng.*, 2: 94-97.
- Zhang, J., X. Cai and G. Liu, 2008b. The role of aspects in software product lines. *Proceedings of the International Conference on Computer Science and Information Technology*, Aug. 29-Sept. 2, Singapore, pp: 588-592.