Scientific Research Publishing

Journal of Software Engineering and Applications, 2019, 12, 293-306
http://www.scirp.org/journal/jsea
ISSN Online: 1945-3124
ISSN Print: 1945-3116

Investigation of Tools, Techniques and Languages for Model Driven Software Product Lines (SPL): A Systematic Literature Review

Nazish Yousaf[1,2], Maham Akram[2], Amna Bhatti[2], Ammara Zaib[2]

[1]Department of Computer & Software Engineering, College of Electrical and Mechanical Engineering (NUST), Islamabad, Pakistan
[2]Department of Computer Science, University of Wah, Wah Cantt, Pakistan
Email: nazish.yousaf15@ce.ceme.edu.pk, mahamakram55@gmail.com, amnabhatti070@gmail.com, ammara.zaib5@gmail.com

How to cite this paper: Yousaf, N., Akram, M., Bhatti, A. and Zaib, A. (2019) Investigation of Tools, Techniques and Languages for Model Driven Software Product Lines (SPL): A Systematic Literature Review. *Journal of Software Engineering and Applications*, **12**, 293-306.
https://doi.org/10.4236/jsea.2019.127018

Received: April 4, 2019
Accepted: July 28, 2019
Published: July 31, 2019

Abstract

Software development is a complex and difficult task that requires the investment of sufficient resources and carries major risk of failure. Model Driven Engineering (MDE) focuses on creating software models and automating code generation from the models. Model Driven Software Development (MDSD) offers significantly more effective approaches. These approaches improve the way of building software. Model driven approaches partially increase developer productivity, decrease the cost of software construction, improve software reusability, and make software more maintainable. This paper investigates the methods where Model Driven Software Development is integrated with Software Product Line (SPL). This SLR has been conducted to identify 71 research works published since 2014. We have collected 18 tools, 14 techniques and 17 languages used for MDSD for SPL. We analyze which technique is suitable for SPL. We compare the techniques on the basis of features provided by these tools to understand the better-quality results.

Keywords

Model Driven SPL, Model Based Product Lines, Modeling for SPL, Model Development for SPL

1. Introduction

When design changes to a software product lines, developer faced with uncertainty about deciding among multiple possible SPL designs. As the requirement of the system changes according to environment, the SPL grows day by day. Software Product Lines (SPLs) are families of related software systems developed for specific market segments or domains [1] [2]. SPL is used for reusability of resources in family of products [3] [4]. In the competitive market, the most challenging task in SPL is designing the architecture. To manage the common and variable features of SPL, a tree structure known as Feature Model (FM), is commonly used in literature. Feature Model is also the tools and techniques used for creating a collection of similar software systems from a shared set of software assets [5] [6]. Each product of SPL differs from the others with variable features that provide functionalities according to end user requirements and specifications [7]. Developers use SPL to increase the reusability of features that reduce the development cost and time to market, which results in better product development. SPL is an approach used to develop a range of software products with a high degree of similarity. In this approach, a feature model is usually used to keep track of similarities and differences [8]. The latest generation of SPL methods and tools are demonstrating even greater improvements and extending more benefits. In current software, traceability, productivity and variability play a key role to maintain the consistency between requirement and code generation. Current SPL traceability mechanism trace from variability in feature of variations (feature model, variable model) in configuration of Product Line Architecture (PLA) in term of adding and removing components [9] [10].

The Software Product Line Engineering (SPLE) and Model Driven Engineering (MDE) are two main aspects of software reuse (two software development paradigms) [25]. SPL is a useful and very successful approach to reuse many domains. Two frameworks of SPL are domain engineering and application engineering. In the production of new software system, the domain engineering (also called product line engineering) is used to reuse domain knowledge. It is used to improve and enhance the quality of developed software from the reuse software domain [10]. Application engineering is related with design, management and execution of the system. SPLE is concerned with producing the similar software products. Whereas, MDE focused on creating and modeling the software products. The paper presents an overview of several SPL dimension (*i.e.* SPLE, MDE) to collect the appropriate number of researches.

Model driven SPL is a combination of SPLE and MDE methodologies. In model-driven SPL, instead of creating and modeling the similar software product models individually, the product models are derived from the product line model. Researches have proposed "Model Driven" approaches to improve the state of the art and skills in software engineering. The purpose of this paper is to answer the following research questions in order to summarize MDSPL languages, tools and techniques.

Research Question 1: What are the significant languages used for Model Driven Software Product Lines?

Research Question 2: What are the significant tools used for Model Driven Software Product Lines?

Research Question 3: Which significant techniques are used for Model Driven Software Product Lines?

2. Research Methodology

In this section, we define comprehensive guidelines for selection of research articles. To make our SLR more comprehensive and result-oriented we have selected different research papers published from 2014 to 2019, from top leading databases. Keywords used for search are "model-driven SPL", "model-based product lines", "modeling for SPL", and "model development for SPL". Thus, collected studies helped us to establish more accurate and reliable results on the subject. All the research articles considered in this study were published in one of the following top research databases.

We have selected 71 research publications from different databases published from 2014 to 2019, as shown in Table 1. We have selected five scientific databases *i.e.* "IEEE", "SPRINGER", "ACM", "ELSEVIER", and "WILEY" for collection of significant researches and the detailed search process is given in Figure 1.

3. Analysis and Results

3.1. MDSPL Languages Identified

We have identified 17 languages which are being used for Model-driven SPL as shown in Table 2. And mostly languages are used in software engineering like UML, Domain Specific Modelling Languages (DSML), Common Variability Language (CVL), SysML, Atlas Transformation Language (ATL), Variability Modeling Language (VML), Xtext and Xtend2, Object-Oriented Language (OOL), and QVT (Query/View/Transformation). These languages are used by many developers to get a required output. These languages are helpful in designing and coding phase.

3.2. MDSPL Tools Identified

We have identified 18 tools are used in this paper but most frequently used tools are, SPLOT Prototyping, Visio, Kola Maker, SuperMod, Protégé tools and INCOSE as shown in Table 3. All identified tools are used to construct models. These tools provide a wide variety of build in shapes, objects, symbols to work with it.

Table 1. Research publication selection details.

Database	Type	Selected research works	No. of researches
IEEE	J	[2] [3] [8] [17] [19] [26] [27] [30]	8
	C	[1] [4] [5] [7] [11] [12] [13] [14] [15] [18] [20]-[25] [31] [33]	18
SPRINGER	J	[6] [9] [10] [16] [28] [29] [32] [34]	8
ACM	J	[55] [56] [57] [58] [59]	5
	C	[35]-[54]	20
ELSEVIER	J	[60]-[66]	7
WILEY	C	[67] [68] [69] [70] [71]	5
		Total	71

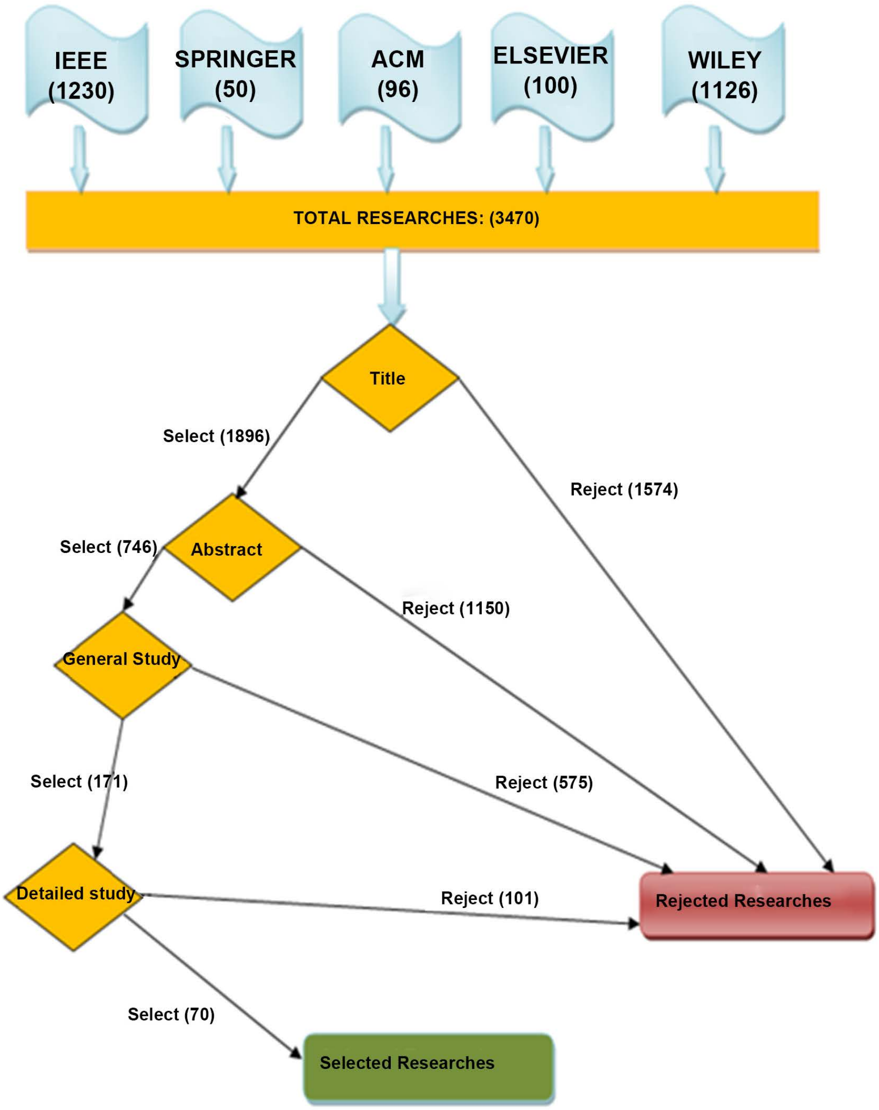

Figure 1. Search process.

3.3. MDSPL Techniques Identified

We have identified 14 techniques used for MDSPL as shown in Table 4. Mostly used techniques are Feature Model (FM), Variable Model (VM), Product line architecture (PLA), Orthogonal Variability Model (OVM), Feature-oriented domain analysis (FODA) method, and Feature-oriented reuse method (FORM). These techniques are used widely nowadays for better quality products. MDA uses these techniques to construct models.

4. Answers to Research Questions

***Research Question* 1:** What are the significant languages used for Model Driven Software Product Lines?

Answer: Use of 16 languages has been identified by our research papers as shown in Table 2. UML is widely used language as it is the backbone of all

Table 2. Identified MDSPL languages.

Sr. No.	MDSPL languages	No. of researches	Research identifications
1	UML	21	[1] [5] [12] [17] [24] [25] [26] [28] [29] [31] [33] [34] [41] [51] [53] [54] [55] [56] [61] [66] [68]
2	Domain Specific Modelling Languages (DSMLs)	11	[8] [10] [17] [18] [20] [21] [26] [35] [53] [55] [66]
3	SysML	4	[29] [46] [61] [68]
4	Common Variability Language (CVL)	3	[3] [26] [20]
5	Atlas Transformation Language (ATL)	3	[4] [25] [18]
6	Variability Modeling Language (VML)	2	[21] [33]
7	Xtext and Xtend	2	[4] [55]
8	Object-oriented language (OOL)	2	[30] [34]
9	QVT (Query/View/Transformation)	2	[4] [18]
10	XML based language	1	[17]
11	Alloy Clafer	1	[22]
12	Ontology web language (OWL)	1	[23]
13	DeltaJ	1	[6]
14	VSL (variability specification language)	1	[17]
15	Variability Instantiation Language (VIL) and Variability Template Language(VTL)	1	[35]
16	Business Processes Modelling Language (BPML)	1	[13]
17	OPM model	1	[70]

languages. It visualized, specify and construct the document of the software. DSML is used mostly to develop Software, Application and Domain and design the models. OOP is used not to show only the data structure but also data type.

***Research Question* 2:** What are the significant tools used for Model Driven Software Product Lines?

Answer: There are 17 tools identified from selected 71 papers as shown in Table 3. SPLOT identify the similarities and differentiation among software product. It is used for editing, configuring, and debugging FM easily. The basic task is software product line is the analysis of feature model. Analysis tasks are checking the validity and counting valid configurations. SPLOT also contains world's largest repository of feature models, open model adopted allowing anyone to create or share model with SPL.

Table 3. Identified MDSPL tools.

Sr. No	MDSPL tools	No. of researches	Research identifications
1	SPLOT (Software Product Lines Online Tools) tool	7	[1] [3] [11] [17] [19] [26] [49] [67]
2	Visio	2	[1] [26]
3	Koala Maker	2	[1] [26]
4	SuperMod	2	[31] [42]
5	Protégé tools	2	[23] [28]
6	ZipMe	1	[25]
7	Rational Software Architect (RSA)	1	[5]
8	DOORS and Rhapsody	1	[29]
9	Horizontal and vertical tools	1	[34]
10	EUD (End User Development)	1	[37]
11	IMITATOR	1	[45]
12	PTC Integrity Modeler tool.	1	[46]
13	Combinatorial Test Design (CTD)	1	[52]
14	FINDBUGS	1	[57]
15	xMapper and xLineMapper	1	[58]
16	RED-PL tool	1	[61]
17	Ohai	1	[65]

***Research Question* 3:** Which significant techniques are used for Model Driven Software Product Lines?

Answer: We have identified 14 techniques in our research paper work as shown in Table 4. Software product line development uses different techniques and models. Feature model is one of the main techniques used for MDSPL. Feature model is a compact representation of all the products of the SPL in term of features. Feature model is visually represented by means of a feature diagram.

5. Discussion and Limitations

There are some important languages that are identified in most of the researches as shown in Table 2. Unified modeling language (UML) models can be annotated with performance properties. We defined a comprehensive classification of behavioral variability in behavioral models including UML. UML plays an important role in modeling, classifying, visualizing and constructing the documents. It is a general-purpose modeling language in field of software engineering that provides a standard way to visualize the design of a system and also help in modeling the system. Architecture Description Language (ADL) formally used to alleviating the software development complexity and making this development process with less error rate. It represents formal notations for describing software architectures in terms of coarse-grained components and connectors.

Table 4. Identified MDSPL techniques.

Sr. No	MDSPL techniques	No. of researches	Research identifications
1	Feature Model (FM)	33	[1] [2] [3] [5] [6] [7] [10] [11] [12] [14] [15] [17] [19] [22] [24] [25] [28] [29] [31] [33] [36] [37] [38] [39] [45] [47] [49] [50] [51] [56] [60] [63] [67] [69]
2	Variable Model (VM)	18	[3] [4] [13] [21] [23] [26] [31] [33] [35] [36] [38] [41] [42] [43] [52] [53] [55] [64]
3	PLA (Product line architecture)	4	[9] [29] [39] [58]
4	Orthogonal Variability Model (OVM)	4	[19] [46] [56] [61]
5	Feature-oriented domain analysis (FODA) method	5	[10] [18] [65] [69] [71]
6	Feature oriented reuse method (FORM)	3	[18] [65] [71]
7	Model-based testing (MBT)	1	[19]
8	Reduction technique	1	[63]
9	Aspect oriented (AO) Technique	1	[39]
10	Component-based development (CBD) and aspect-oriented development (AOD) Technique	1	[40]
11	Ontology-based technique	1	[15]
12	Mapping	1	[4]
13	Traditional software metric methods	1	[32]
14	Latent Semantic Analysis (LSA)	1	[20]

Koala is also an ADL as it supports the description of the structure of a configuration in terms of its components.

The most commonly used technique we found is the feature modeling. It is widely used notation as well as important technique that describes the set of products in SPL. FM is used as Domain requirement variability modeling, which describes the group of requirements such as increased productivity, handling defects and improving quality. It is also widely used during whole product line development process. Feature Modeling is an important technique used to manage common and variable features of Software Product Line (SPL) in applications such as Internet of Things (IoT). It usually used to keep track of similarities and differences. But there are also some complexity issues that the developers face. Moreover, it is very time-consuming task. In recent research, there are scalability issues in SPL product due to occurrence of hardens and constraint violation between relationships of features in final product development. Likewise, variability modeling (VM) aims at creating, evolving, and configuring variability models, which describe the common and variable characteristics.

We briefly describe the techniques along with features as shown in **Table 5**. This table labels the techniques that we identify from selected researches. The

Table 5. Comparison of techniques based on features.

Techniques	Features				
	Productivity	Variability	Reusability	Reliability	Traceability
Feature Model (FM)		√	√	√	
Variable Model (VM)					√
Orthogonal Variability Model (OVM)	√				
Feature-oriented domain analysis (FODA) method		√			
Feature oriented reuse method (FORM)	√				
Product Line Architecture (PLA)	√			√	√
Reduction technique		√			
Aspect oriented (AO) Technique			√		
Component-based development (CBD) and aspect-oriented development (AOD) Technique			√		
Ontology-based technique			√		
Mapping			√		√
Traditional software metric methods			√	√	
Latent Semantic Analysis (LSA)					√

purpose of this table is to assure that these techniques have important features that make ease for developer to choose according to their requirements needs. Feature model and PLA are more reliable as it has almost all features that show in below table. SPL requires more variability and reusability for software application.

We attempted to observe SPL guidelines, however, still there are some limitations in our work. We have selected a restricted set of keywords and there is a chance that we might have left some important keywords. We have used filter of 2014-19 for selection of research article work, we might have missed some important papers published before 2014. In the same way, we have only chosen five main databases for our research article selection, we might have lost some significant researches from other scientific database engines.

6. Conclusions

This SLR studies the detailed analysis of SPL integrated with MDSD for modern application and domain. We have identified 17 languages, 18 tools and 14 techniques from selected 71 research studies published since 2014. The paper also provides a detailed comparison between the identified techniques on the basis of features provided by them. We concluded that SPL is not bounded to any specific rule, techniques or tool. SPL can make work easy not complex for any application. Model-Driven techniques are more enhanced and maintainable so that it increases the customer satisfaction.

In future work, we intend to implement Automated SPL Testing by integrating model driven testing with SPL with wide-ranging details.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Eason, G., Noble, B. and Sneddon, I.N. (1955) On Certain Integrals of Lipschitz-Hankel Type Involving Products of Bessel Functions. *Philosophical Transactions of the Royal Society of London. Series A*, **247**, 529-551. https://doi.org/10.1098/rsta.1955.0005

[2] Cohan, A.Z., Coha, A.Z., Bibi, A. and Motla, Y.Z. (2017) Optimized Software Product Line Architecture and Feature Modeling in Improvement of SPL. *International Conference on Frontiers of Information Technology*, Islamabad, 18-20 December 2017, 167-172.

[3] Abbas, A., Siddiqui, I.F., Lee, S.U.J., *et al.* (2018) Multi-Objective Optimum Solutions for IoT-Based Feature Models of Software Product Line. *Journal of Latex Class Files*, **6**, 12228-12239. https://doi.org/10.1109/ACCESS.2018.2806944

[4] Burger, E., Henss, J., Küster, M. and Kruse, S. (2016) View-Based Model-Driven Software Development with Model Join. *Journal of Software and Systems Modeling*, **15**, 473-496. https://doi.org/10.1007/s10270-014-0413-5

[5] Wang, S., Ali, S., Yue, T. and Liaaen, M. (2013) Using Feature Model to Support Model-Based Testing of Product Lines: An Industrial Case Study. 13*th International Conference on Quality Software*, Nanjing, 29-30 July 2013, 75-84. https://doi.org/10.1109/QSIC.2013.51

[6] Damiani, F., Faitelson, D. and Gladisch, C. (2017) A Novel Model-Based Testing Approach for Software Product Lines. *Journal of Software and Systems Modeling*, **16**, 1223-1251. https://doi.org/10.1007/s10270-016-0516-2

[7] Abbas, A., Siddiqui, I.A., Lee, S.U.J. and Bashir, A.K. (2017) Binary Pattern for Nested Cardinality Constraints for Software Product Line of IoT-Based Feature Models. *Journal of Latex Class Files*, **5**, 3971-3980. https://doi.org/10.1109/ACCESS.2017.2680470

[8] Akbar, M.A., Sang, J., Khan, A.A., *et al.* (2016) Improving the Quality of Software Development Process by Introducing a New Methodology-AZ-Model. *IEEE Access*, **6**, 4811-4823. https://doi.org/10.1109/ACCESS.2017.2787981

[9] Schwagerl, F., Buchmann, T. and Westfechtel, B. (2015) SuperMod—A Model-Driven Tool That Combines Version Control and Software Product Line Engineering. 10*th International Joint Conference on Software Technologies*, Vol. 2, Colmar, 20-22 July 2015, 1-14. https://doi.org/10.5220/0005506600050018

[10] Diaz, J., Pérez, J. and Garbajosa, J. (2015) A Model for Tracing Variability from Features to Product-Line Architectures: A Case Study in Smart Grids. *Journal of Requirements Engineering*, **20**, 323-323. https://doi.org/10.1007/s00766-014-0203-1

[11] Tanhaei, M. and Habibi, J. (2016) A Feature Model Based Framework for Refactoring Software Product Line Architecture. *Journal of Computer Science and Technology*, **2**, 951-986. https://doi.org/10.1007/s11390-016-1674-y

[12] Taentzer, G., Salay, R., Strüber, D. and Chechick, M. (2017) Transformations of Software Product Lines: A Generalizing Framework Based on Category Theory. *ACM/IEEE* 20*th International Conference on Model Driven Engineering Languages and Systems*, Austin, 17-22 September 2017, 101-111. https://doi.org/10.1109/MODELS.2017.22

[13] Shofi, I.M. and Fajar, A.N. (2017) Conceptual Approach for Gathering SPL Requirement from Goal Model. 5*th International Conference on Cyber and IT Service Management*, Denpasar, 8-10 August 2017, 1-5. https://doi.org/10.1109/CITSM.2017.8089308

[14] Oliveira, A.L.D., Braga, R.T., Masiero, P.C., Papadopoulos, Y., Habli, I. and Kelly, T. (2014) A Model-Based Approach to Support the Automatic Safety Analysis of Multiple Product Line Products. *Brazilian Symposium on Computing Systems Engineering*, Manaus, 3-7 November 2014, 7-12. https://doi.org/10.1109/SBESC.2014.20

[15] Nie, L., Zhang, L. and Geng, Z. (2012) Product Line Variability Modeling Based on Model Difference and Merge. *IEEE* 36*th Annual Computer Software and Applications Conference Workshops*, Izmir, 16-20 July 2012, 509-513. https://doi.org/10.1109/COMPSACW.2012.95

[16] Schwagerl, F. and Numao, M. (2014) A Product Domain Model Based Software Product Line Engineering for Web Application. 2*nd International Symposium on Computing Networking*, Shizuoka, 10-12 December 2014, 572-576.

[17] Khan, F.Q. and Musa, S. (2017) A Study: Selection of Model Metamodel and SPL Tools for the Verification of Software Product Lines. *International Journal of Information Technology*, **9**, 353-362. https://doi.org/10.1007/s41870-017-0055-2

[18] Guo, J., Wang, Y., Zhang, Z., Nummenmaa, J. and Niu, N. (2012) Model-Driven Approach to Developing Domain Functional Requirements in Software Product Lines. *Journal of IET Software*, **6**, 391-401. https://doi.org/10.1049/iet-sen.2010.0072

[19] Gebizi, C.S. and Sözer, H. (2016) Model-Based Software Product Line Testing by Coupling Feature Models with Hierarchical Markov Chain Usage Models. *IEEE International Conference on Software Quality, Reliability and Security Companion*, Vienna, 1-3 August 2016. https://doi.org/10.1109/QRS-C.2016.42

[20] Font, J., Arcega, L., Haugen, O. and Cetina, C. (2018) Achieving Feature Location in Families of Models through the Use of Search-Based Software Engineering. *Journal of Transactions on Evolutionary Computation*, **22**, 363-377. https://doi.org/10.1109/TEVC.2017.2751100

[21] Filho, J.B.F., Barais, O., Baudry, B. and Noir, J.L. (2012) Leveraging Variability Modeling for Multi-Dimensional Model-Driven Software Product Lines. 3*rd International Workshop on Product Line Approaches in Software Engineering*, Zurich, 4 June 2012, 5-8. https://doi.org/10.1109/PLEASE.2012.6229774

[22] Famelis, M., Rubin, J., Czarnecki, K., Salay, R. and Chechik, M. (2017) Software Product Lines with Design Choices: Reasoning about Variability and Design Uncertainty. 20*th International Conference on Model Driven Engineering Languages and Systems*, Austin, 17-22 September 2017, 93-100. https://doi.org/10.1109/MODELS.2017.3

[23] Fajar, A.N. and Shofi, I.M. (2016) Development of SPL Government System with Ontology Web Language. 4*th International Conference on Cyber and IT Service Management*, Bandung, 26-27 April 2016, 1-4. https://doi.org/10.1109/CITSM.2016.7577587

[24] Schwagerl, F., Buchmann, T., Uhrig, S. and Westfechtel, B. (2015) Towards the Integration of Model-Driven Engineering, Software Product Line Engineering, and Software Configuration Management. 3*rd International Conference on Mod-*

el-Driven Engineering and Software Development, Angers, 9-11 February 2015, 1-14. https://doi.org/10.1007/978-3-319-27869-8_2

[25] Assunção, W.K.G. (2015) Search-Based Migration of Model Variants to Software Product Line Architectures. *IIEEE/ACM 37th IEEE International Conference on Software Engineering*, Florence, 16-24 May 2015, Vol. 2, 895-898. https://doi.org/10.1109/ICSE.2015.286

[26] Dageforde, J.C., Reischmann, T., Majchrzak, T.A. and Ernsting, J. (2016) Generating App Product Lines in a Model-Driven Cross-Platform Development Approach. *49th Hawaii International Conference on System Sciences*, Koloa, 5-8 January 2016, 5803-5812. https://doi.org/10.1109/HICSS.2016.718

[27] Tawhid, R. and Petriu, D.C. (2011) Product Model Derivation by Model Transformation in Software Product Lines. *14th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops*, Newport Beach, 28-31 March 2011, 72-79.

[28] Duran-Limon, H.A., Garcia-Rios, C.A., Castillo-Barrera, F.E. and Capilla, R. (2015) An Ontology-Based Product Architecture Derivation Approach. *Journal of Transactions on Software Engineering*, **41**, 1153-1168. https://doi.org/10.1109/TSE.2015.2449854

[29] Derakhshanmanesh, M., Fox, J. and Ebert, J. (2014) Requirements-Driven Incremental Adoption of Variability Management Techniques and Tools: An Industrial Experience Report. *Journal of Requirements Engineering*, **19**, 333-354. https://doi.org/10.1007/s00766-013-0185-4

[30] Cu, C. and Zheng, Y. (2016) Architecture-Centric Derivation of Product in a Software Product Line. *IEEE/ACM 8th International Workshop on Modeling in Software Engineering*, Austin, 14-22 May 2016, 27-33. https://doi.org/10.1145/2896982.2896988

[31] Asteasuain, F. and Vultaggio, L.P. (2016) Declarative and Flexible Modeling of Software Product Line Architectures. *IEEE Latin America Transactions*, **14**, 885-892. https://doi.org/10.1109/TLA.2016.7437236

[32] Alferez, M., Bonifácio, R. and Teixeira, L. (2014) Evaluating Scenario-Based SPL Requirements Approaches: The Case for Modularity, Stability and Expressiveness. *Journal of Requirements Engineering*, **19**, 355-376. https://doi.org/10.1007/s00766-013-0184-5

[33] Zhao, F., Nian, G., Jin, H., Yang, L.T. and Zhu, Y. (2017) A Hybrid eBusiness Software Metrics Framework for Decision Making in Cloud Computing Environment. *IEEE Systems Journal*, **11**, 1049-1059. https://doi.org/10.1109/JSYST.2015.2443049

[34] Ntanos, E. and Dimitriou, G. (2018) A Model-Driven Software Engineering Workflow and Tool Architecture for Servitised Manufacturing. *Journal of Information Systems and e-Business Management*, **16**, 683-720. https://doi.org/10.1007/s10257-018-0371-5

[35] Schmid, K., Kröher, C. and El-Sharkawy, S. (2018) Model-Based Product Line Development with Easy-Producer Using VIL and VTL. *Proceedings of the 22nd International Systems and Software Product Line*, Gothenburg, 10-14 September 2018, Vol. 1, 303. https://doi.org/10.1145/3233027.3233058

[36] Beuche, D. (2017) Using Feature Models to Manage Variability and Requirements Reuse. *Proceedings of the 21st International Systems and Software Product Line*, Sevilla, 25-29 September 2017, 249. https://doi.org/10.1145/3106195.3106229

[37] Tezermes, V. and Gomaa, H. (2015) A Software Product Line Approach for End User Development of Smart Spaces. *Proceedings of the 5th International Workshop*

on Product Line Approaches in Software Engineering, Florence, 19 May 2015. https://doi.org/10.1109/PLEASE.2015.14

[38] Sampaio, G., Borba, P. and Teixeira, L. (2016) Partially Safe Evolution of Software Product Lines. *Proceedings of the* 20*th International Systems and Software Product Line Conference*, Beijing, 16-23 September 2016, 124-133. https://doi.org/10.1145/2934466.2934482

[39] Tan, L. and Lin, Y. (2015) An Aspect-Oriented Feature Modelling Framework for Software Product Line Engineering. *Proceedings of the* 24*th Australasian Software Engineering Conference*, Adelaide, 28 September-1 October 2015, 111-115. https://doi.org/10.1145/2811681.2811703

[40] Azzolini, R.P., Rubira, C.M.F., Tizzei, L.P., Gaia, F.N. and Montecchi, L. (2015) Evolving a Software Products Line for E-Commerce Systems: A Case Study. *Proceedings of the* 2015 *European Conference on Software Architecture Workshops*, Dubrovnik, 7-11 September 2015, Article No. 26. https://doi.org/10.1145/2797433.2797460

[41] Gaeta, J.P. and Czarnecki, K. (2015) Modeling Aerospace Systems Product Lines in SysML. *Proceedings of the* 19*th International Conference on Software Product Line*, Nashville, 20-24 July 2015, 293-302. https://doi.org/10.1145/2791060.2791104

[42] Schwagerl, F. and Westfechtel, B. (2016) SuperMod: Tool Support for Collaborative Filtered Model-Driven Software Product Line Engineering. *Proceedings of the* 31*st IEEE/ACM International Conference on Automated Software Engineering*, Singapore, 3-7 September 2016, 822-827. https://doi.org/10.1145/2970276.2970288

[43] Lity, S., Nahrendorf, S., Thum, T., Seidl, C. and Schaefer, I. (2018) 175 % Modeling for Product-Line Evolution of Domain Artifacts. *Proceedings of the* 12*th International Workshop on Variability Modelling of Software-Intensive Systems*, Madrid, 7-9 February 2018, 27-34. https://doi.org/10.1145/3168365.3168369

[44] Campos, J.C., Fayollas, C., Goncalves, M., Martinie, C., Navarre, D., Palanque, P. and Pinto, M. (2017) A More Intelligent Test Case Generation Approach through Task Models Manipulation. *Proceedings of the ACM on Human-Computer Interaction Archive*, **1**, Article No. 9. https://doi.org/10.1145/3095811

[45] Luthmann, L., Stephan, A., Bürdek, J. and Hau, M.L. (2017) Modeling and Testing Product Lines with Unbounded Parametric Real-Time Constraints. *Proceedings of the* 21*st International Systems and Software Product Line*, Sevilla, 25-29 September 2017, 104-113. https://doi.org/10.1145/3106195.3106204

[46] Bilic, C., Sundmark, D., Afzal, W., Wallin, P., Causevic, A. and Amlinger, C. (2018) Model-Based Product Line Engineering in an Industrial Automotive Context: An Exploratory Case Study. *Proceedings of the* 22*nd International Systems and Software Product Line*, Gothenburg, 10-14 September 2018, Vol. 2, 56-63. https://doi.org/10.1145/3236405.3237200

[47] Gregg, S.P., Scharadin, R. and Clements, P. (2015) The More You Do, the More You Save: The Superlinear Cost Avoidance Effect of Systems Product Line Engineering. *Proceedings of the* 19*th International Conference on Software Product Line*, Nashville, 20-24 July 2015, 303-310. https://doi.org/10.1145/2791060.2791065

[48] Lity, S., Kowal, M. and Schaefer, I. (2016) Higher-Order Delta Modeling for Software Product Line Evolution. *Proceedings of the* 7*th International Workshop on Feature-Oriented Software Development*, Amsterdam, 30 October 2016, 39-48. https://doi.org/10.1145/3001867.3001872

[49] Pereira, J.A., Matuszyk, P., Krieter, S., Spiliopoulou, M. and Saake, G. (2016) A Feature-Based Personalized Recommender System for Product-Line Configuration. *ACM SIGPLAN Notices*, **52**, 120-131. https://doi.org/10.1145/3093335.2993249

[50] Kuhn, T., Kassin, K.I., Cazzola, W. and Abmann, U. (2018) Modular Feature-Oriented Graphical Editor Product Lines. *Proceedings of the 22nd International Systems and Software Product Line*, Gothenburg, 10-14 September 2018, Vol. 1, 76-86. https://doi.org/10.1145/3233027.3233034

[51] Anthonysamy, P. and Somé, S.S. (2008) Aspect-Oriented Use Case Modeling for Software Product Lines. *Proceedings of the AOSD Workshop on Early Aspects*, Brussels, 31 March 2008, Article No. 5. https://doi.org/10.1145/1404946.1404951

[52] Kang, S., Kim, J., Baek, H., Ahn, H., Jung, P. and Lee, J. (2017) Comparison of Software Product Line Test Derivation Methods from the Reuse Viewpoint. *Proceedings of the 6th International Conference on Software and Computer Applications*, Bangkok, 26-28 February 2017, 1-8. https://doi.org/10.1145/3056662.3056663

[53] Schwagerl, F. and Westfechtel, B. (2017) Perspectives on Combining Model-Driven Engineering, Software Product Line Engineering, and Version Control. *Proceedings of the 11th International Workshop on Variability Modelling of Software-Intensive Systems*, Eindhoven, 1-3 February 2017, 76-83. https://doi.org/10.1145/3023956.3023969

[54] Pietsch, C., Reuling, D., Kelter, U. and Kehrer, T. (2017) A Tool Environment for Quality Assurance of Delta-Oriented Model-Based SPLs. *Proceedings of the 11th International Workshop on Variability Modelling of Software-Intensive Systems*, Eindhoven, 1-3 February 2017, 84-91. https://doi.org/10.1145/3023956.3023960

[55] Acher, M., Lopez-Herrejon, R.E. and Rabiser, R. (2017) Teaching Software Product Lines: A Snapshot of Current Practices and Challenges. *22nd International Systems and Software Product Line Conference*, Gothenburg, 10-14 September 2018, 249. https://doi.org/10.1145/3233027.3236393

[56] Bashroush, R., Garba, M., Rabiser, R., Groher, I. and Botterweck, G. (2017) CASE Tool Support for Variability Management in Software Product Lines. *Journal of Computing Surveys*, **50**, 1-45. https://doi.org/10.1145/3034827

[57] Thum, T., Apel, S., Kästner, C., Schaefer, I. and Saake, G. (2014) A Classification and Survey of Analysis Strategies for Software Product Lines. *Journal of Computing Surveys*, **47**, Article No. 6. https://doi.org/10.1145/2580950

[58] Behiati, R., Nejati, S. and Briand, L.C. (2014) Architecture-Level Configuration of Large-Scale Embedded Software Systems. *Journal of Software Engineering and Methodology*, **23**, Article No. 25. https://doi.org/10.1145/2581376

[59] Zheng, Y., Cu, C. and Taylor, R.N. (2018) Maintaining Architecture-Implementation Conformance to Support Architecture Centrality: From Single System to Product Line Development. *Journal of Software Engineering and Methodology*, 27, Article No. 8. https://doi.org/10.1145/3229048

[60] Rincon, L., Giraldo, G., Mazo, R., Salinesi, C. and Diaz, D. (2015) Method to Identify Corrections of Defects on Product Line Models. *Journal of Computer Science*, **314**, 61-81. https://doi.org/10.1016/j.entcs.2015.05.005

[61] Hause, M. and Korff, A. (2016) Decision-Driven Product Development. *INCOSE International Symposium*, **26**, 2446-2461. https://doi.org/10.1002/j.2334-5837.2016.00305.x

[62] Sepulveda, S., Cravero, A. and Cachero, C. (2016) Requirements Modeling Languages for Software Product Lines: A Systematic Literature Review. *Journal of Information and Software Technology*, **69**, 16-36. https://doi.org/10.1016/j.infsof.2015.08.007

[63] Sabouri, H. and Khosravi, R. (2014) Reducing the Verification Cost of Evolving Product Families Using Static Analysis Techniques. *Journal of Science of Computer*

Programming, **83**, 35-55. https://doi.org/10.1016/j.scico.2013.06.009

[64] Galindo, J.A., Dhungana, D., Rabiser, R., Benavides, D. and Grünbacher, P. (2015) Supporting Distributed Product Configuration by Integrating Heterogeneous Variability Modeling Approaches. *Journal of Information and Software Technology*, **62**, 78-100. https://doi.org/10.1016/j.infsof.2015.02.002

[65] Camacho, C., Llana, L. and Núñez, A. (2016) Cost-Related Interface for Software Product Lines. *Journal of Logical and Algebraic Methods in Programming*, **85**, 227-244. https://doi.org/10.1016/j.jlamp.2015.09.009

[66] Beohar, H., Varshosaz, M. and Mousavi, M.R. (2016) Basic Behavioral Models for Software Product Lines: Expressiveness and Testing Pre-Orders. *Journal of Science of Computer Programming*, **123**, 42-60. https://doi.org/10.1016/j.scico.2015.06.005

[67] Bhushan, M., Goel, S. and Kumar, A. (2017) Improving Quality of Software Product Line by Analyzing Inconsistencies Models Using an Ontological Rule-Based Approach. *Journal of Expert Systems*, **35**, e12256. https://doi.org/10.1111/exsy.12256

[68] Hause, M. and Hummell, J. (2015) Model-Based Product Line Engineering Enabling Product Families with Variants. *IEEE Aerospace Conference*, Big Sky, 7-14 March 2015, 25. https://doi.org/10.1002/j.2334-5837.2015.00132.x

[69] Pereira, J.A., Maciel, L. and Figueiredo, T.F.N.E. (2017) Heuristic and Exact Algorithms for Product Configuration in Software Product Lines. *Proceedings of the* 22*nd International Systems and Software Product Line*, Gothenburg, 10-14 September 2018, Vol. 1, 247-247. https://doi.org/10.1145/3233027.3236395

[70] Sharon, A. and Dori, D. (2017) Model-Based Project-Product Lifecycle Management and Gantt Chart Models: A Comparative Study. *Systems Engineering*, **20**, 447-466. https://doi.org/10.1002/sys.21407

[71] Mannion, M. and Kaindl, H. (2014) Engineering Requirements in Product Lines. *INCOSE International Symposium*, Denver, 23-23 June 2011, Vol. 21, 3139-3215. https://doi.org/10.1002/j.2334-5837.2011.tb01308.x