Requirements Engineering and Variability Management in DSPLs Domain Engineering: A Systematic Literature Review

Léuson M. P. da Silva[1], Carla I. M. Bezerra[1,2,3], Rossana M. C. Andrade[2,3] and José Maria S. Monteiro[2]

[1]*Federal University of Ceará (UFC), Quixadá Campus, ZIP: 63902-580, Quixadá, CE, Brazil*

[2]*Federal University of Ceará (UFC), Pici Campus, ZIP: 60455-760, Fortaleza, CE, Brazil*

[3]*Group of Computer Networks, Software Engineering and Systems (GREat), Pici Campus, Bloco 942-A, ZIP: 60455-760, Fortaleza, CE, Brazil*

Keywords: Dynamic Software Product Line, Requirements Engineering, Variability Management.

Abstract: Recently, Software Product Lines (SPLs) have been used successfully for building products families. However, the currently and complex software products demand more adaptive features. Today, many application domains demand capabilities for flexible adaptation and post-deployment reconfiguration. In this context, Dynamic Software Product Lines (DSPLs) represent a way to produce software products able to change their own behavior at runtime due to the changes in the product use environment. DSPLs present some interesting properties such dynamic variability and reconfiguration at runtime. The dynamic variability is represented by the definition of variants and context information. The reconfiguration at runtime is the process that enables the features activation and deactivation in a configuration product. Both properties are closely related to the requirements engineering and variability management, in the domain engineering life-cycle. In this research, we provide a systematic literature review that aims to identify the activities, assets, tools and approaches that are used in requirements engineering and variability management in DSPLs domain engineering. We performed a manual and automatic search, resulting in 581 papers of which 37 were selected. We also provide a discussion about the challenges and solutions of runtime variability mechanisms in the context of DSPLs.

1 INTRODUCTION

Software Product Lines (SPLs) can be defined as a set of software-intensive systems sharing a common and managed set of features that satisfies the specific needs of a particular market segment or mission (Northrop et al., 2007). However, SPLs just support the development of static products (Hinchey et al., 2012), i.e., SPLs products are not able to adapt their own behavior to the changes in the users needs at runtime (Bencomo et al., 2012). On the other hand, the currently complex systems need to deal with dynamic aspects, such as successive reconfigurations at runtime, after their first deployment (Bosch et al., 2015). In this context, emerged Dynamic Software Product Lines (DSPL) (Hallsteinsen et al., 2008).

DSPLs extend existing product line engineering approaches by moving their capabilities to runtime (Hinchey et al., 2012). In DSPLs, products can be reconfigured dynamically at runtime after their initial derivation (Bencomo et al., 2012). Although DSPLs have some differences compared with SPLs, DSPLs still share the same development life-cycle as presented by Capilla et al. (Capilla et al., 2014a).

DSPLs, as well SPLs, are composed of two main development life-cycle: domain and application engineering (Hallsteinsen et al., 2008). The domain engineering is responsible for (i) specifying, documenting and developing the assets that will be used to compose the future products of the line. Besides it is responsible for (ii) producing the necessary SPLs infrastructure, composed of: a common architecture and its variation points, a set of reusable parts and a model to represent the variability (Bencomo et al., 2012).

Once DSPLs adapt their own behavior at runtime, besides to identify the requirements, it is necessary to recognize the contexts that the line will need to support. These tasks are performed in the domain engineering life-cycle, through two different activities: domain and context analysis (Capilla et al., 2014a). The domain analysis specifies the domain that the line will support, identifying and documenting the vari-

Silva, L., Bezerra, C., Andrade, R. and Monteiro, J.
Requirements Engineering and Variability Management in DSPLs Domain Engineering: A Systematic Literature Review.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 544-551
ISBN: 978-989-758-187-8

able features of the domain (Capilla et al., 2014b). The context analysis captures the contexts to be supported by the DSPL (Capilla et al., 2014b). An important activity from context analysis is to identify the information used by the products reconfiguration process that happens due to the changes in use environment. When a new context is identified, the product needs to check which features must be activated and which should not (Capilla et al., 2014a).

Guedes et al. (Guedes et al., 2015) present a systematic mapping focused on DSPLs aspects to identify methodologies that are used to execute the variability management. However, the results do not present what activities are used to project and how these activities need to be executed to ensure the DSPLs variability was well understood. In the work of Da Silva et al. (da Silva et al., 2013) is presented a SLR that aims at understanding how dynamic derivation is made in DSPL. The work identifies how the models, approaches and methods are used to address the dynamic derivation problem in DSPLs, but it is not presented how these assets are made, what information, roles and activities are involved to define them.

In this context, we performed a Systematic Literature Review (SLR) to investigate how the requirements engineering and variability management are performed in DSPLs domain engineering. We aimed to identify the activities, assets, tools and approaches that are used. As result, we analyzed 37 studies dated from 2008 to 2015. The main contribution of this work is a catalogue of activities to support the requirements process in DSPLs domain engineering.

The remainder of this work is organized as follows. Section 2 reports the SLR. Section 3 describes the studies classification. The results of each research questions are presented in section 4. Section 5 presents a discussion about the results. Section 6 discusses the threats of validity. Section 7 concludes this work and presents suggestions for future work.

2 SYSTEMATIC REVIEW PROCESS

The review process of this work followed the guidelines of (Keele, 2007) and (Kitchenham et al., 2009). The process included the definition of three activities: planning, conducting and results reporting. In the planning was defined the review protocol and in the conducting, the focus was on the selection and analyses process of the work. Finally, the results reporting comprised the results presentation.

To support the SLRs execution we use a tool to automatize some steps. The adopted tool was StArt [1] (State of the Art through Systematic Reviews) that supports the review process since the definition of the review protocol until the results report. Due to StArt be a desktop tool, some steps were supported by templates allowing the parallel work among the researchers. The next subsections present how the review process was done.

2.1 Research Questions

This work followed a main research question and six (6) secondary questions. The first four secondary questions are related to list the activities, assets, tools and the approaches used to represent the variability and requirements in DSPLs domain engineering. Additionally, the last two research questions deal with the DSPLs variability and aim to specify what information are used, still in the requirements engineering, to do or just to support, the variability management at runtime.

- RQ1. How are the requirements engineering (RE) and variability management (VM) executed in DSPLs domain engineering?
  - RQ1.1. What activities of RE and VM are used in DSPLs?
  - RQ1.2. What approaches are used to document the requirements in DSPLs?
  - RQ1.3. What assets are built of RE and VM in DSPLs?
  - RQ1.4. What approaches are used to represent the DSPLs variability?
  - RQ1.5. What approaches are used to support the reconfiguration process in DSPLs?
  - RQ1.6. What approaches are used to eliminate possible inconsistencies in DSPLs variability model?

2.2 Search Process

The search process started with a manual and automated search, in digital libraries (DL). The adopted DLs were: Scopus [2], Compendex[3] and Web of Science[4]. The manual search was necessary due to the verification that the automated search did not return papers from important conferences dated from 2015 (SPLC, VaMoS). After the two searches, all identified papers were joined at the same work set.

[1] http://lapes.dc.ufscar.br/tools/
[2] http://www.scopus.com/
[3] http://www.engineeringvillage.com/
[4] https://apps.webofknowledge.com/

To perform the automated search, we used a search string. This string was defined through keywords, extracted from each research question. At the end, the selected words were joined with search operators (AND, OR). To ensure the string was appropriated to return valid papers, we tested it many times. The adopted search string is presented bellow:

("Software product line engineering" OR "Domain Engineering" OR "Domain Analysis" OR "Context Analysis") AND (Requirements OR "Requirements Engineering" OR Elicitation OR Analysis OR Specification OR Verification OR Management) AND ("feature model" OR "variability model" OR "decision model" OR "domain model") AND ("Software Product Line" OR "product family") AND (autonomic OR pervasive OR ecosystems OR dynamic OR "context-aware" OR adaptive)*

2.3 Inclusion/Exclusion and Quality Criteria

The papers of the automated and manual search were analyzed according to some criteria. To justify the reason that a paper would be selected or not, we determine inclusion and exclusion criteria. Additionally, the papers content should be evaluated. It was done following the guidelines of Kitchenham et al. (Kitchenham et al., 2006). For that, we define 6 questions and a valuation function. The evaluation score could vary from 0, for papers that do not satisfy a criterion, to 6, for papers that satisfy totally a criterion. The inclusion/exclusion criteria are listed in Table 1.

Table 1: Inclusion/Exclusion criteria.

Inc/Exc	Criterion
Inc.	Activities about RE and VM for DSPLs domain engineering
Inc.	Approaches for the RE and VM activities
Inc.	Approaches to support the requirements and variability
Inc.	Approaches to support the inconsistencies treatment in DSPLs variability model
Exc.	Do not focus on RE and VM of DSPLs
Exc.	Not written in English
Exc.	Out of the valid formats (papers from conferences and journals)
Exc.	Could not be accessed from UFC's network or by contacting authors

2.4 Data Extraction

The adopted approach for data extraction was based on the work of Montagud et al. (Montagud et al., 2012). For each research question was defined some values that could be the answers presented by the papers. It was done to make easy the extraction process. The information to be extracted were: activities from domain and context analysis, extracted from Capilla et al. (Capilla et al., 2014b), involved roles, built assets, variability representation, inconsistency treatment, validation type and related adopted method, and context use.

3 STUDIES CLASSIFICATION

The studies classification process corresponds to the conducting and review reporting phases of the SLR. This process was done supported by four researchers and made in 5 steps:

- Step 1: Perform string search on DLs and the manual search;
- Step 2: Elimination of duplicated papers using the StArt tool;
- Step 3: Title, Abstract and Keyword analysis of all papers according to the inclusion/exclusion criteria;
- Step 4: Full reading and analysis of the papers according to the inclusion/exclusion criteria by the researchers through pairs read;
- Step 5: Last check for duplicated papers identification, and information extraction and quality evaluation execution.

If during the classification process of a paper a nonconformity among the researchers happens, the researchers are responsible for solving the conflict and decide together if the paper would be eliminated or not. An overview of this process is presented in Figure 1. It is possible to see the number of returned papers of each source and the number of selected papers after each step, respectively. Finally, after the step five, the set of papers decreased to 37 papers that were analysed again to extract the principal information.

4 RESULTS

This section presents the SLR results from the 37 selected work. The papers of our work were selected in August/2015. Due to it some papers from important

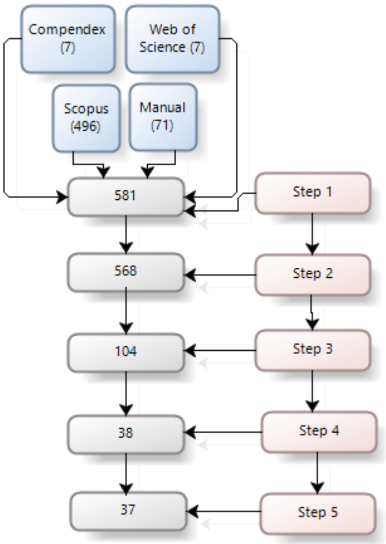

Figure 1: Study selection process.

sources dated after august are not presented here. The next subsections present each adopted research question and their related results.

4.1 Papers Overview

Before the analysis of each research question is important to present an overview about the selected papers. Figure 2 presents the number of selected papers along the years according to the three selection process. According to Figure 2 is possible to see that the oldest selected work in the 3rd selection dates from 2008. This means that the researches in DSPLs is still a new research area. From 2011 the number of selected papers per year is more regular varying between 4 and 7 papers.

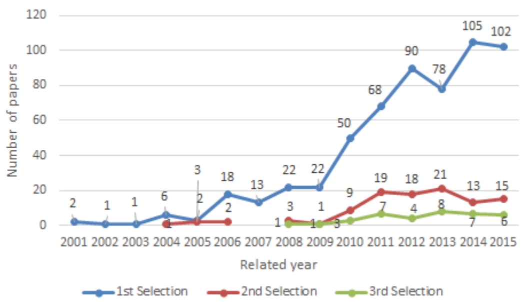

Figure 2: Selected papers along the years.

About the papers use context, the majority (29) was developed to academic ends, while 8 papers were focused on the industrial area. It reinforces the idea of DSPL is a new research area, but there are some work that promote its use in industry. About the quality evaluation of the papers only one has the maximal score 6 (S07), while the majority has a score between 3 and 5. Table 6 presents the the quality evaluation of each paper.

4.2 Requirements Engineering and Variability Management in DSPLs Domain Engineering (RQ1)

The main research question aims to know how the requirements engineering and variability management are executed in DSPLs domain engineering. The results of each secondary question are presented as follows.

4.2.1 RQ1.1 What Activities of RE and VM are used in DSPLs?

Table 2 shows the identified activities. As can be seen, the activities are separated in three groups (phases). The first group brings the activities commonly used in the traditional software development, domain analysis. The second group presents the activities of the context analysis. Some papers present the activity Define operational rules of the domain analysis phase that is related to the architecture modelling. This kind of activity is commonly executed on the project domain phase, like some papers execute. This activity is attributed to the the second group, when it is executed on the domain analysis phase, and attributed to the third group, when a paper performs this activity on the project domain phase. A finding is that the activity Define multiple connection is executed only in the project domain phase.

Although the papers present specific activities to attend to the domain specification, some details about these activities execution were clearer in the section of studies cases. However, the identified activities are more related to requirements and variability modeling, and how the variability management is done at design time and runtime. Activities about requirements elicitation are executed with fewer importance.

This question identified too the roles involved in the activities, shown in Table 2. Just 9 papers present the roles involved in their activities. Although the roles are more involved in the project specification and modelling, there is not a clear definition about the responsibilities of each one. For example, the designers and the architects are responsible for modeling the DSPLs architecture, S05 and S09. The analysts are involved in the activities of modeling such as identifying of features, S06 e S07. On the other hand, the definition of the developer role just combines the responsibilities of the other roles. It difficulties the understanding about which roles are necessary to ex-

Table 2: Result from the first research question.

Domain Engineering			
	Domain Analysis		Domain Project
	Domain Analysis	Context Analysis	Domain Project
Activities	Conception, Elicitation, Specification, Validation	Identify physical properties, Identify context features, Model context features, Define operational rules	Define operational rules, Define multiple connection
Roles	Analyst (S06, S07), Designer (S05, S06, S15, S16, S26, S35), Architect (S09), Developer (S31)		

ecute each activity.

The identified activities are supported by some tools. Most papers (64%) did not use tools to support their process or they did not mention it. The rest of the papers use tools to project modelling as also to verify the consistency and correctness of the models. Table 3 presents the tools related to each purpose.

Table 3: Tools support.

Purpose	Tools and Papers
Workflow representation	BPMN (S10)
Variability modelling	Fama (S02), MOSkitt4SPL (S07), Atlas Model (S07), BPMN (S10), Familiar (S26), FeatureIDE (S26), eMoflon (S8), Odyssey (S28), DOPLER (S33), VariaMos (S36, S37)
Verification models	GNU Prolog (S07), Clafer (S10), Familiar (S26), VariaMos (S36, S37).

4.2.2 RQ1.2 What Approaches are used to Document the Requirements in DSPLs?

Only 6 papers present approaches that are used to document the requirements. The adopted approaches vary of the use UML diagrams, class diagram and use case (S13), sequence diagram (S15), approach that does not support the concept of variability, until the use of new approaches like Schemas (S25) and Goals (S35). S01 and S05 organize the domain requirements in feature model, but they do not present what approaches are used to transform the requirements, textual specification, in features of the variability model.

4.2.3 RQ1.3 What Assets are Built of RE and VM in DSPLs?

Table 4 presents all identified assets. The feature model as also the context feature model represent the assets with the higher frequency, 37% each one. The feature model is used to represent the DSPLs variability, as is used by traditional SPLs. The deference in DSPLs is about the context feature model that joins the context features, necessary to the reconfiguration process. The next more used model is the extended feature model (6 papers), this model extends the feature model to support specific needs.

Table 4: Built Assets.

Assets	Papers
Aspect model	S23, S31
Base composition model and Composition model	S07
Base Model	S10
Context feature model	S01, S06, S07, S10, S11, S12, S20, S22, S28, S29, S30, S32, S34, S37
Extended feature model	S05, S08, S10, S16, S18, S34
Feature model	S01, S02, S03, S04, S07, S12, S13, S20, S21, S26, S27, S28, S32, S37
Feature model adaptation	S01
Requirements specification	S13, S15, S25, S35
Rules adaptions	S24
States machine	S08
Weaving model	S07, S10

4.2.4 RQ1.4 What Approaches are used to Represent the DSPLs Variability?

The models used to represent the variability were: Aspect model (S31), Actor model (S17), MVRP (S16), OWL (S12), OCL (S04) and Feature Model (S01, S02, S03, S07, S10, S11, S13, S14, S15, S19, S21, S23, S26, S27, S28, S29, S34, S37).

The feature model represents the most used approach to represent the variability due to its use flexibility. This approach allows to change its own properties in order to attend the new use needs. Other consideration about the feature model is that there is not a pattern to represent the variability. For example, some papers identify context features but the way to organize these information vary. Some put the context features at the same feature model used to represent the variability while others put it in a independent model.

4.2.5 RQ1.5 What Approaches are used to Support the Reconfiguration Process in DSPLs?

This question is related to the approaches that are used to support the adaptability at runtime. This activity is more related to the domain design. Table 5 presents the results of this question.

Table 5: Approaches to reconfiguration process.

Approaches	Studies
Adaptation rules	S07, S24
ECA	S01, S02, S22, S27, S31, S32
MAPE-K	S15, S36
MAPE	S01, S35
Transformation rules	S26
Aspect models	S23
Context mapping	S10
PrtNets	S30
Contraint-satisfaction	S21

ECA (Event-Condition-Action) approach is the most widely used approach. ECA is based on rules that are created from constraints of different sources, such as the activity responsible for identifying operational rules (see subsection 4.2.1). The process of specifying the adaptation rules through MAPE (Monitor, Analyze, Plan and Execute) and MAPE-K (Monitor, Analyze, Plan, Execute and Knowledge) follows almost the same methodology for both. It is necessary to identify how the context information would be accessible, specifying the constraints that would be responsible to decide to what new context the product would change and what parts would be necessary to support this new context.

4.2.6 RQ1.6 What Approaches are used to Eliminate Possible Inconsistencies in DSPLs Variability Model?

In DSPLs it is necessary to ensure that not only the variability model at design time is consistent as also the consistency of the model when a reconfiguration happens. Although this consistency checking be an important step to ensure and validate the requirements, only 12 papers mentioned it. Eight papers (S07, S08, S10, S20, S22, S26, S28, S37) executed checks just to ensure the adopted variability model did not have inconsistencies. It was used tools to execute this checking (the list of identified tools is presented in Table 3).

Related to the papers that did checking at runtime (S02, S05, S06, S08, S15), only specific papers (e.g., S08, S15) present the approaches they adopted. The checking process followed the use of adaptation rules and the application of algorithms that are responsible for evaluating if a new configuration has any inconsistencies or not. The other papers just cite they did it but they did not present details about.

5 DISCUSSION

The domain engineering is responsible for exploring the domain that the DSPL will support. The results of this work show that DSPLs research has produced a set of assets, tools and approaches to support the activities necessary for DSPL requirements engineering and variability management.

About the activities of domain analysis, we concluded that the activities responsible for the domain specification and modeling have more importance that the others, like conception and elicitation. Because of it most papers do not specify formally the requirements domain. It goes against the domain analysis goal. The same problem happens with the activities of context analysis. The activities focus on the contexts identification and modeling, while the activities that treats the rules definition, responsible for identification of changes in a context, receives less importance. Still about domain analysis, we identified that the papers do not present activities to support the requirements changes as also do not present how the changes are treated, when they happen.

A challenge about the reconfiguration process is to change the variability model at runtime when a reconfiguration happens. Tools are used to check the model structure and its consistence but it was not identified yet a tool that supports this verification at runtime. Another important finding is about how the non-functional requirements (NFR) are identified and treated in domain engineering. Only one paper (S15) treated this issue with the same importance that functional requirements (FR) receive. The NFRs are more related to the domain project, that is when the architecture is modeled. Although this relation between the architecture modeling and the NFRs, it is important that the activity of architecture modeling receives the necessary information about the NFRs from domain analysis instead to identify the motivations around them. Other finding about NFRs is how the reconfiguration process is done. The NFRs are not considered by the papers in this process.

S09 is interested in evaluating the DSPLs quality attributes but they just focused on the features model. Evaluating the quality attributes in all DSPLs domain engineering using others assets represents a new research opportunity.

6 THREATS TO VALIDITY

This section discusses threats that could have affected the SLR results. We verified the threats were about the construct validity. Construct validity is concerned whether the treatments reflect the cause and the outcome reflects the effect (Wohlin et al., 2012).

The first possible threat is about the search string. It needs to reflect the main objective of the study otherwise would be returned work out of the study area. We have tried to minimize it through successive searches in DLs. As result, we verified that always some important papers, that we knew before, were returned (S06, S13). The second threat might have been about the search process. We verified papers from important conferences dated from 2015 were not been returned by the DLs. To solve this, we did a manual search in the proceedings of these conferences

7 CONCLUSIONS AND FUTURE WORK

This SLR aimed to determine how the requirements engineering and variability management supports the DSPLs domain engineering. To answer this question, we identified relevant and current studies following a formal approach. These studies were analyzed, evaluated and the information were extracted.

The results show the activities are concentrated on DSPLs modeling and specification. Traditional approaches (UML diagrams) can be used to document the domain requirements as also the feature model, that can be also used to represent the domain variability. The assets built in the activities can be done using tools responsible for modeling and consistence verification of them. We identified the following research opportunities: modeling and treatment of NFRs still in domain analysis; evaluating quality attributes using assets different of variability models; a mechanism to check the consistence and that be able to modify the variability model at runtime; defining a pattern to represent the variability in DSPLs; and exploring approaches that can be used for eliciting the FRs.

As future work, we would like to determine approaches to support the found gaps and to define a formal process for DSPLs RE and VM.

REFERENCES

Bencomo, N., Hallsteinsen, S., and Santana de Almeida, E. (2012). A view of the dynamic software product line landscape. *Computer*, 45(10):36–41.

Bosch, J., Capilla, R., and Hilliard, R. (2015). Trends in systems and software variability. *IEEE Software*, (3):44–51.

Capilla, R., Bosch, J., Trinidad, P., Ruiz-Cortés, A., and Hinchey, M. (2014a). An overview of dynamic software product line architectures and techniques: Observations from research and industry. *Journal of Systems and Software*, 91:3–23.

Capilla, R., Ortiz, O., and Hinchey, M. (2014b). Context variability for context-aware systems. *Computer*, (2):85–87.

da Silva, J. R. F., Pereira da Silva, F. A., do Nascimento, L. M., Martins, D. A., and Garcia, V. C. (2013). The dynamic aspects of product derivation in dspl: A systematic literature review. In *Information Reuse and Integration (IRI), 2013 IEEE 14th International Conference on*, pages 466–473. IEEE.

Guedes, G., Silva, C., Soares, M., and Castro, J. (2015). Variability management in dynamic software product lines: A systematic mapping. In *Components, Architectures and Reuse Software (SBCARS), 2015 IX Brazilian Symposium on*, pages 90–99. IEEE.

Hallsteinsen, Svein andHinchey, M., Park, S., and Schmid, K. (2008). Dynamic software product lines. *Computer*, 41(4):93–95.

Hinchey, M., Park, S., and Schmid, K. (2012). Building dynamic software product lines. *Computer*, (10):22–26.

Keele, S. (2007). Guidelines for performing systematic literature reviews in software engineering. In *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*.

Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., and Linkman, S. (2009). Systematic literature reviews in software engineering–a systematic literature review. *Information and software technology*, 51(1):7–15.

Kitchenham, B., Mendes, E., and Travassos, G. H. (2006). A systematic review of cross-vs. within-company cost estimation studies. In *Proceedings of the 10th international conference on Evaluation and Assessment in Software Engineering*, pages 81–90. British Computer Society.

Montagud, S., Abrahão, S., and Insfran, E. (2012). A systematic review of quality attributes and measures for software product lines. *Software Quality Journal*, 20(3-4):425–486.

Northrop, L., Clements, P., Bachmann, F., Bergey, J., Chastek, G., Cohen, S., Donohoe, P., Jones, L., Krut, R., Little, R., et al. (2007). A framework for software product line practice, version 5.0. *SEI.–2007–http://www. sei. cmu. edu/productlines/index. html*.

APPENDIX

Table 6: Quality evaluation of the studies.

ID	Reference	Quality Score
S01	L. Shen, X. Peng, and W. Zhao. Software product line engineering for developing self-adaptive systems: Towards the domain requirements. In Computer Software and Applications Conference (COMPSAC), 2012 IEEE 36th Annual.	4,25
S02	A. S. Nascimento, C. M. Rubira, and F. Castor. Arcmape: A software product line infrastructure to support fault-tolerant composite services. In High-Assurance Systems Engineering (HASE), 2014 IEEE 15th International Symposium.	3,25
S03	D. Dermeval, T. Tenorio, I. I. Bittencourt, A. Silva, S. Isotani, and M. Ribeiro. Ontology-based feature modeling: An empirical study in changing scenarios. Expert Systems with Applications, 42(11):4950–4964, 2015.	3,75
S04	C. Dubslaff, C. Baier, and S. Kluppelholz. Probabilistic model checking for feature-oriented systems. In Transactions on Aspect-Oriented Software Development XII, pages 180–220. Springer, 2015.	3,5
S05	R. Mizouni, M. A. Matar, Z. Al Mahmoud, S. Alzahmi, and A. Salah. A framework for context-aware self-adaptive mobile applications spl. Expert Systems with applications, 41(16):7549–7564, 2014.	5,25
S06	R. Capilla, J. Bosch, P. Trinidad, A. Ruiz-Cortes, and M. Hinchey. An overview of dynamic software product line architectures and techniques: Observations from research and industry. Journal of Systems and Software, 91:3–23, 2014.	4,5
S07	G. H. Alferez, V. Pelechano, R. Mazo, C. Salinesi, and D. Diaz. Dynamic adaptation of service compositions with variability models. Journal of Systems and Software, 91:24–47, 2014.	6
S08	J. Burdek, S. Lity, M. Lochau, M. Berens, U. Goltz, and A. Schurr. Staged configuration of dynamic software product lines with complex binding time constraints. In Proceedings of the Eighth International Workshop on Variability Modelling of Software-Intensive Systems, page 16. ACM, 2014.	4,25
S09	L. E. Sánchez, J. A. Diaz-Pace, A. Zunino, S. Moisan, and J.-P. Rigault. An approach for managing quality attributes at runtime using feature models. In Software Components, Architectures and Reuse (SBCARS), 2014 Eighth Brazilian Symposium on, pages 11–20.	3,75
S10	A. Murguzur, X. De Carlos, S. Trujillo, and G. Sagardui. Context-aware staged configuration of process variants@ runtime. In Advanced Information Systems Engineering, pages 241–255. Springer, 2014.	4,75
S11	K. Saller, M. Lochau, and I. Reimund. Context-aware dspls: model-based runtime adaptation for resource-constrained systems. In Proceedings of the 17th International Software Product Line Conference co-located workshops, ACM, 2013.	3,75
S12	C. Cetina, P. Giner, J. Fons, and V. Pelechano. Prototyping dynamic software product lines to evaluate run-time reconfigurations. Science of Computer Programming, 78(12):2399–2413, 2013.	4,25
S13	F. G. Marinho, R. M. Andrade, C. Werner, W. Viana, M. E. Maia, L. S. Rocha, E. Teixeira, J. B. Ferreira Filho, V. L. Dantas, F. Lima, et al. Mobiline: A nested software product line for the domain of mobile and context-aware applications. Science of Computer Programming, 78(12):2381–2398, 2013.	4,25
S14	I. Kumara, J. Han, A. Colman, T. Nguyen, and M. Kapuruge. Sharing with a difference: realizing service-based saas applications with runtime sharing and variation in dynamic software product lines. In Services Computing (SCC), 2013 IEEE International Conference on, pages 567–574.	3,25
S15	C. Ghezzi and A. M. Sharifloo. Dealing with non-functional requirements for adaptive systems via dynamic software product-lines. In Software Engineering for Self-Adaptive Systems II, pages 191–213. Springer, 2013.	5,25
S16	L. Jean-Baptiste, S. Maria-Teresa, G. Jean-Marie, and B. Antoine. Modeling dynamic adaptations using augmented feature models. In Proceedings of the 28th Annual ACM Symposium on Applied Computing, pages 1734–1741. ACM, 2013.	4,25
S17	H. Sabouri and R. Khosravi. Modeling and verification of reconfigurable actor families. JUCS, 19(2):207–232, 2013.	3,25
S18	D. Kramer, C. Sauer, and T. Roth-Berghofer. Towards explanation generation using feature models in software product lines. Knowledge Engineering and Software Engineering (KESE), page 13, 2013.	4,25
S19	V. T. Sarinho, A. L. Apolinario, and E. S. de Almeida. Oofm-a feature modeling approach to implement mpls and dspls. In 2012 IEEE 13th International Conference on Information Reuse & Integration (IRI), 2012.	2,25
S20	F. G. Marinho, P. H. Maia, R. Andrade, V. M. Vidal, P. A. Costa, and C. Werner. Safe adaptation in context-aware feature models. In Proceedings of the 4th International Workshop on Feature-Oriented Software Development, ACM, 2012.	4,25
S21	C. Parra, D. Romero, S. Mosser, R. Rouvoy, L. Duchien, and L. Seinturier. Using constraint-based optimization and variability to support continuous self-adaptation. Proceedings of the 27th ACM Symposium on Applied Computing, 2012.	3
S22	F. G. Marinho, R. Andrade, and C. Werner. A verification mechanism of feature models for mobile and context-aware software product lines. In Software Components, Architectures and Reuse (SBCARS), 2011 Fifth Brazilian Symposium on.	4,25
S23	C. Parra, X. Blanc, A. Cleve, and L. Duchien. Unifying design and runtime software adaptation using aspect models. Science of Computer Programming, 76(12):1247–1260, 2011.	3,75
S24	M. Rosenmuller, N. Siegmund, M. Pukall, and S. Apel. Tailoring dynamic software product lines. In ACM SIGPLAN Notices, volume 47, pages 3–12. ACM, 2011.	3,25
S25	J. Dehlinger and R. R. Lutz. Gaia–pl: a product line engineering approach for efficiently designing multiagent systems. ACM Transactions on Software Engineering and Methodology (TOSEM), 20(4):17, 2011.	4
S26	M. Acher, P. Collet, P. Lahire, S. Moisan, and J.-P. Rigault. Modeling variability from requirements to runtime. In Engineering of Complex Computer Systems (ICECCS), 2011 16th IEEE International Conference on, pages 77–86.	4,5
S27	L. Shen, X. Peng, J. Liu, and W. Zhao. Towards feature-oriented variability reconfiguration in dynamic software product lines. In Top Productivity through Software Reuse, pages 52–68. Springer, 2011.	3,75
S28	P. Fernandes, C. Werner, and E. Teixeira. An approach for feature modeling of context-aware software product line. J. UCS, 17(5):807–829, 2011.	4
S29	Z. Jaroucheh, X. Liu, and S. Smith. Mapping features to context information: Supporting context variability for context–aware pervasive applications. In Web Intelligence and Intelligent Agent Technology, 2010, International Conference on.	3,75
S30	Z. Jaroucheh, X. Liu, and S. Smith. Candel: product line based dynamic context management for pervasive applications. In Complex, Intelligent and Software Intensive Systems (CISIS), 2010 International Conference on, pages 209–216.	3
S31	T. Dinkelaker, R. Mitschke, K. Fetzer, and M. Mezini. A dynamic software product line approach using aspect models at runtime. In 5th Domain-Specific Aspect Languages Workshop, 2010.	4,5
S32	M. Acher, P. Collet, F. Fleurey, P. Lahire, S. Moisan, and J.-P. Rigault. Modeling context and dynamic adaptations with feature models. In 4th International Workshop Models@ run. time at Models 2009 (MRT'09), page 10, 2009.	3,25
S33	R. Wolfinger, S. Reiter, D. Dhungana, P. Grunbacher, and H. Prahofer. Supporting runtime system adaptation through product line engineering and plug-in techniques. In Composition-Based Software Systems, 2008, Seventh International.	3
S34	R. Capilla, M. Hinchey, and F. J. Daz. Collaborative context features for critical systems. In Proceedings of the Ninth International Workshop on Variability Modelling of Software-intensive Systems, page 43. ACM, 2015.	3,5
S35	J. C. Muñoz-Fernández, G. Tamura, I. Raicu, R. Mazo, and C. Salinesi. Refas: a ple approach for simulation of self-adaptive systems requirements. Proceedings of the 19th International Conference on Software Product Line, ACM, 2015.	4,25
S36	N. Abbas and J. Andersson. Harnessing variability in product-lines of self-adaptive software systems. In Proceedings of the 19th International Conference on Software Product Line, pages 191–200. ACM, 2015.	2,25
S37	R. Mazo, J. C. Muñoz-Fernández, L. Rincon, C. Salinesi, and G. Tamura. Variamos: an extensible tool for engineering (dynamic) product lines. Proceedings of the 19th International Conference on Software Product Line. ACM, 2015.	3,5