Scandic Journal of Advanced Research and Reviews

ISSN: 2703-965X CODEN (USA): SJARCA
Cross Ref DOI: dx.doi.org/10.55966/sjarr
Journal homepage: www.sjarr.com

Review on product derivation approaches in the software product line

Rana Fezan Afzal[1] and Khalid Mahmood[2]

1, MS scholar, International Islamic University Islamabad, Pakistan
ranafezan.afzal@gmail.com

2, PhD scholar, Doctor of Philosophy in Computer Science (Ph.D.) Continue Hitec University, Session Taxila, Pakistan

Scandic Journal of Advanced Research and Reviews, 2022, 2(02), 042–056

Article DOI: https://doi.org/10.55966/sjarr.2022.2.2.0033

Abstract: Product derivation is one of the most important challenges in the software product line. Obtaining the individual product from the shared software is a costly and time-consuming job. Various approaches have been proposed for product derivation in the software product line in previous times. This paper will review approaches concerning the product derivation in the software product line. This paper will provide a state-of-the-art literature review on product derivation in software line approaches. Moreover, this will be more useful in order to obtain a novel valid feather combination approach for product derivation in software product line.

Introduction

A software product line is a group of software-intensive systems that share frequently organized features designed from a common set of main resources in a prearranged means [1].

The main thing is the set of shared assets that combine all artifacts. These common assets are utilized on sharing based not only for separate usage.

There are two main areas in the software product line: the domain and application engineering. Core assets are associated with domain engineering, which is based on the standard and varying feathers in the products. Application engineering is the development of individual products related to reusing the core assets specifically for each product.

Suppose we discuss the product derivation developed with those artifacts belonging to the family of shared products. The term used as product derivation is related to making products from product family software resources [2].

The purpose of adopting the software product line is to increase the business's productivity by reducing the production cost and minimizing the time in bringing the product to the market. This encourages product derivation approaches in the software product line[5].

Product derivation is main thing in the application engineering that address the development of the product from the core assets of the software product line. Hence its said that the prodct derivation is the process of making th epriduct sing the assest of the product software line family. Here we are sharing the image thet will describes the software product line engineering framework.

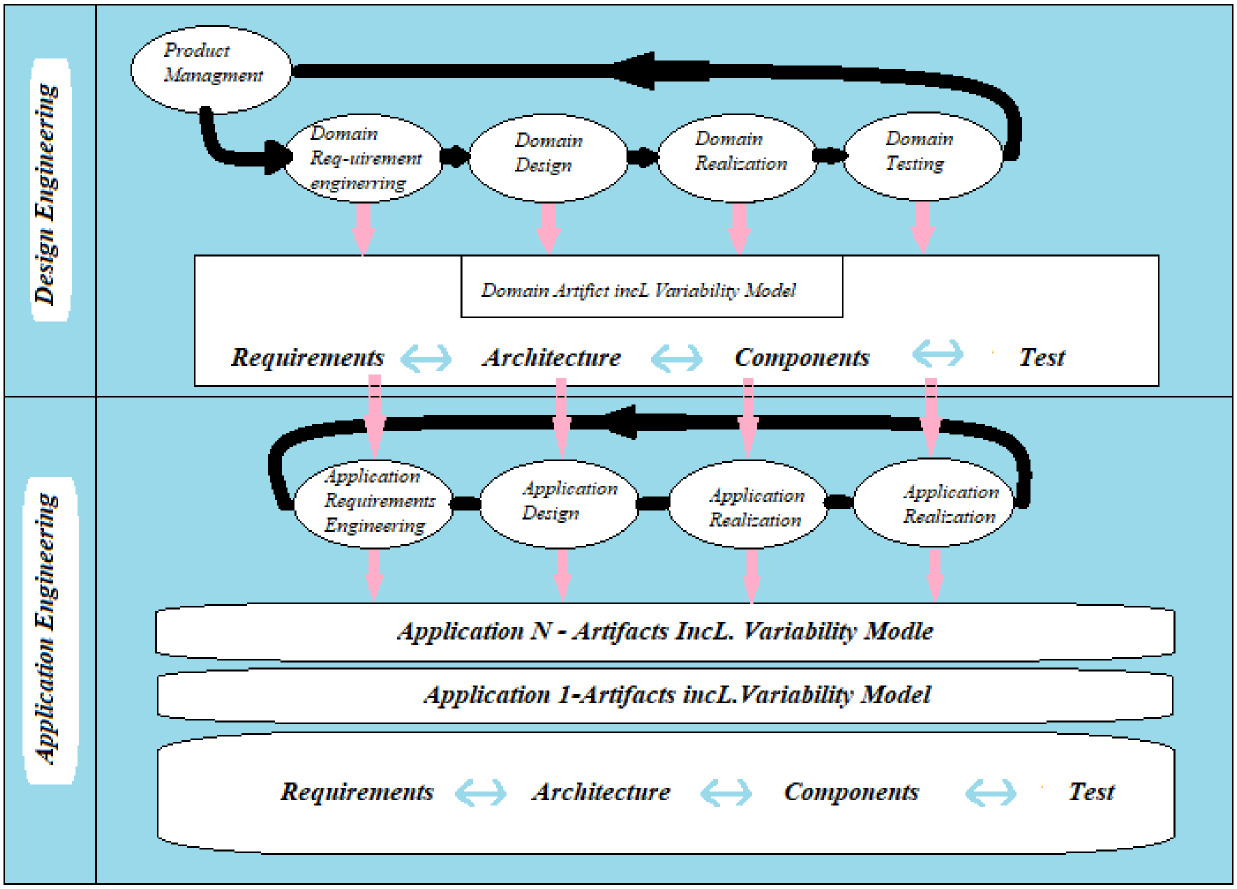

Figure 1: Software product line engineering framework

Now we discuss the main actions for product derivation in software products which are mainly three get ready for derivation, product derivation or configuration, and the additional development and testing. The approaches for product derivation are always initiated from scratch, and the next step will be features selection and decision making. Overall the observation is needed to be considered these activities, which are affirmed and translated client needs, describe the bottom pattern, map client's needs, describe responsibility and job structure, generate derivation guidelines. Now we have to go for the next step of the product derivation second main activity, which was product derivation or configuration. The primary purpose is to make the product by reusing the product artifacts within less time and low cost by ensuring customers' satisfaction. Mainly two things needed to be considered for the product derivation: the assets selection creating partial product creations. After that, the last critical activity is the product deployment and testing, which is the duty of the creators to execute the product and ensure the changes at the product level. The main things concluded for the product testing, and deployment stage is the part makings and checking, component combination, incorporation trying, and the system testing [3].

The basics of the product derivation framework are two-dimensional classification families and generic software derivation. In two-dimensional families, the primary dimension scope of reuse covers the similarities among the product exploited. Four scopes of reuse are standard infrastructure, plate form, software product line configuration product family. The second dimension is domain scope, representing the main domain and the domain in which predicting family is applied. The second dimension is based on four levels single, program, and hierarchal product family and products population. The generic product derivation process is generalized into a single product family. This is based on two phases, initial and iteration. Initial one created from family assets and iteration arrangement is customized in a quantity of succeeding circulations awaiting the product suitably execute the compulsory needs [20]

This review article is structured as follows: Section 2 gives us the background knowledge of proposed approaches concerning the product derivation in the software product line.

Literature Review

In this part of the article, we discuss the approaches for product derivation in software product lines.

[1] This article developer deals with deriving the software product more automatically and efficiently. An approach represents the set of integrated models specific variability and possible

changing options for product variants and the auto running products with changeable models with aspect-oriented techniques. This model is a research prototype which automatically does the important and significant activities. The fundamental reason for this model is to make sure the product derivation is more efficient. AspectJ also has the acceptability with more advance methods more effectively. The AspectJ demonstrates that this model ensures the practice of the more advanced techniques.

[2]People believe that individual products from shared software assets are time-consuming and expensive. That is why lots of efforts concerning the scenario by proposing various approaches for product derivation. This article also proposes research that investigates those issues. They provide the readers with the terminology and concepts concerned with product derivation. Moreover, those issues occurred during the case studies at two heavy industries related to each other, such as comparable or un-mature firms.

[3] This paper presents the two approaches and compares these approaches, which are proposed different projects by different researchers for product derivation in the software product line. Both of these approaches for product derivation independently behave. One is based on the references model, and the second is a tool-supported derivation scheme. The validation gives proof to facilitate the recognized acts are related to product derivation; we conclude that they are measured as a checklist after making or assessing product derivation methods. Moreover this article also gives the ideas about how to organize for derivation form previous approaches. This leads to concentrate on role and tasks for product derivation stakeholders also guldens about creation. Author of this article mainly and strongly focus on interactive nature of product derivation. In results they concludes that product derivation stll an costly activity.

[8] This article describes a model used for product derivation that uses some specifications of known customer needs, features, and artifacts using knowledge. This model is based on a logically based representation langue that interoperates with a defined model supports the activity in the product derivation mechanism by handling dependencies between artifacts. This model makes the knowledge about the artifact tool-based mechanisms that enable them to of product derivations.

In this article another approach discussed that support human developers and not based on automated software working and its representation is in-fact designed for human constantly rather than for machine understanding. In conclusion making knowledge concerning with features and users needs and artifacts are all automatic derivation mechanism.

In [9] is based on a features realization replica base scheme of product derivation features implementation works as middle level among features model and plan completion. This works to capture feature and cross-cutting relations in the better position stage and facilitate making clear the complex mapping among feature and program execution. Thus, the model and role instantiation can enable feature-driven program-level customization and configuration. Feature-Oriented Programming is taken on as the execution technology for product derivation on the program stage. Then program-level work of art can be executed by a characteristic merge to attain the character-driven product derivation finally.

[10]This decision model-based scheme for product derivation creates the software product line using model derivation engineering principles. The main program used is the meta-models, and the model transformation, which are the sets of ordered model transformation rules. It obtains actual software artifacts from primary models and uses the feathers model to express the variability included in the software product line. A transformation program was adopted as per choice for features model due to product line members' changeability to derivate a producer-specific configuration and decision model used to link variants to model transmission rule. Hence, the decision model can adopt a transmission program to derivate products and specific variants.

[11] In this paper, a method identified as KobrA proposed incorporates two paradigms into symmetric combined draw near for software development and maintenance. The main thing for their combination includes support and makes their installation more flexible of the variants of the system and justifies their methodological support for component-based framework development.

[12]. In this paper, the argument about the approaches used to develop producer line existing domains. They proposed a product software engineering scheme to make the conception of deployment of software products line to the great diversity of enterprise field.

To achieve the product's major focal point through the phase of PuLSTM, customization of its parts enhances the introductory abilities maturity, structured evaluations, and adaptations.

[13]This article talks about two models: product line engineering and model-driven structural design, which are very important paradigms that are less costly and efficient to develop the applications. PLE is more effective in the reuse of the common assets, and the model MDA is used to develop an application in diverse platforms with model transformation. Vigilant understanding of product line engineering and model-driven structural design make known that they complement apiece other by overcoming their borders. They define a likelihood of seamless incorporation

follow-on in a complete marketable level scheme. This article, a realistic product line tactic, DREAM proposed taking on main actions of product line engineering and model transformation characteristic of model-driven architecture. Practicing the proposed scheme makes it achievable to professionally and semi-automatically make many applications that differ on the actions and execution stage.

[14] This article shows the critical feature of variability management in SPL for honest representation of variability. Several experiences do at industries about their software mobility model by software companies. Uniform representation of variation position as initial group unit in all concept layers between features and codes, permitting a hierarchical group of the variability. Moreover, allow for primary group illustration of specific, such as one to one and complicated, and n to m, dependency, another one is to permit for making the connections among dependencies. On the hand, variability modeling methods uphold the initial 2 needs, anyhow not having support for remaining. This paper is a structure for variability modeling and COVAMOF, supporting the fundamental needs.

[15]This article discusses the systematic mechanism that gives a structures model for product derivation in the software product line. This model stands on the set of duties, tasks, and artifacts. The proposed model was identified as a process model for product derivation (Pro-PD). This model assesses to compare existing approaches to justify its standard. Pro-PD explains the duties, process artifacts, and roles utilized to obtain products from an SPL. Moreover, its can state thus model was developed for methodological support, and it was repeatedly developed and judged with four research levels. The initial step was evidence-based methodology, which is the key point and sign of this product derivation approach.

[17] This approach is known as the resolution-oriented line engineering for effectual use again: users' main configurations which easy developed at dolppers laboratory the purposed of the automated software engineering. This tool-supported approach was able to adopt variability and assist the sale peoples, non-technicians, and application engineering. This approach is named as DOPLERSucon, based on various activities. The first one is the domain skilled get ready product configuration by making the derivation model. Secondly, Users described in the derivation model carry out the simple product pattern by making a choice noticeable. After that, similarly, they Capture begin product's exact needs. Moreover, developers should carry out the extra expansion

according to the needs. In the end, engineers put together novel growth with the elected and modified products and organize it for the client

[18] This article discusses the software product line and agile practice that emerged new paradigm in the software makings. The primary aim of these approaches is to improve software productivity, minimize time, reduce cost, and enhance customer satisfaction. The writers of these schemes are trying to research the potential of agile approaches for product derivation in SPL. This article gives a sketch of the proposed agile procedure scheme proposed in the paper for product derivation through industry-based case study research.

In this review paper, all of the approaches in various research papers are appropriately referenced. These approaches have been proposed in previous times and recently taken from the published papers.

Table of comparison

Title	Year of publication	Journal	Customer requirements specifications	Key contribution	Expectations in the end
PuLSE-1	1999	Symposium software reusability	Complete project plan	Deriving product from SPL in an efficient and automated way.	Delivery process as per market size
DREAM	2005	IEEE	Translated customers requirements	A pratcial product line method that allows the semi automatically development on large features.	Making of executable application code
KobrA	2009	PhD Thesis. Institute of system engineering and automation in	Use of decisions models	Two approaches based on incorporate the component and software product	Final results must be application, component, context decision with app’s

		John Calper University Linz		line designed for modeling architecture and develop single and family systems.	realization & tree.
COVAMOF	2003	Software variability management workshop. Groningen. The Netherlands.	Bounding the unique identities	COMVAF-VF tool used in this approach. It used to design the variability model for product family. Binds the variation point based on customer requirements.	Configuration of executable product as require.
Pro-PD	2012	Research gate	Translate customers requirements in to organizational language	Develop generic agile process model to identify fundamental mechanism of product derivation	Satisfying product release.
DOPLER^{ucon}	2009	PhD Thesis, Institute for Systems Engineering and Automation, Johannes Kepler University, Linz,	Creation of derivation model based on decryption and purpose and name.	approach for product configuration with capabilities for adapting variability	Installable product delivery.

				models to guide sales people and application developing with PD	
1-AspectJ	2009	Researchgate	Analyzing the existing issues.	Product deriving from SPL with automated and efficient way.	Derivation of automatic executable products
16- A users centered approach	2009	Thesis at Johns capler University linz / JKU	Collaborating with industrial partners and their feedbacks	Support the stakeholders involve in product configuration and help to handle the complexity of product as per applications diversity	Support and help stakeholders
17- Formal approach for product derivation	2008	Researchgate	Concurrency at software and hardware	A real time application development tool from software product line	High performance real time applications development
18-agile process model	2012	Researchgate	Industries base case studies.	This article present the outline of product derivation with agile process	Improve productivity. Reduce product cost and time to market.

Visualization techniques	2007	Lero	Not discussed briefly	Examining the visualization techniques to support the product derivation in SPL.	Support and improve the variability management and product derivation.
22-quality based heuristic-PD approach	2015	Researchgate	Architectural design for product deviation	derive an optimal solution From weights assigned to architectural choices based on this informati on and clients needs.	Enabling the product derivation process from RA
27-Asset based approach	2006	IEEE	Difficulty in selection of desired assets.	This enables the derivation of product theory evaluating the assets in various ways.	How to use asses to makes product.
30-validation of multimodal approach	2011	Springer	Requirements of architectural architecture.	Multi model approach for derivation and imporvements	Quality driven product architecture
2- Features extraction approach	2015	Elsevier	Lacks of knowledge in readers about product derivations	A case study tto inform the users about product derivation mechanis in more effective way.	Providing knowledge to the readers about product derivation

DSL based approach	2016	ACTA INFORMATICA PRAGENSIA	Dealing with those factors which are directly involves with time and cost factors.	DSL based approach integrate the SPL and model driven engineeringto takes thebenifits of both of these.	Modeling the variability model and Development Time reduction

Conclusions

Product derivation is a central issue in software product derivations occurs that deriving the individual products from the shared software assets is time-consuming and costly. It's necessary to develop software with low cost, time-consuming, and more effective and satisfying customers.

This paper presents the approaches reviewed in the literature for product derivation in the software product line. These approaches are proposed by various authors in previous times that have numerous positive and negative aspects. We have discussed then in the paper with short decryption. After this we have made a comparison table of these approaches in various contexts. These approaches are discussed in detail and generate a research question to develop a novel approach for the product derivation in the software product line.

The literature review can perform in the future by including more papers and grouping these approaches based on similarities.

References

[1] Goetz Botterweck, Kwanwoo Lee, Steffen Thiel. Automating Product Derivation in Software Product Line Engineering

[2] Sybren Deelstra, Marco Sinnema *, Jan Bosch. Product derivation in software product families: a case study. ELSEVIER The Journal of Systems and Software 74 (2005) 173–194.

[3] Rick Rabiser, Padriag O leary, Ita Richardson. Key activities for product derivation in software product line.

[4] Jihyun Lee, Sungwon Kang, and Danhyung Lee. A Survey on Software Product Line Testing

[5] John D. McGregor. Preparing for Automated Derivation of Products in a Software Product Line 2005 thesis

[6] Nesrine LAHIANI, Djamal BENNOUAR. A Brief Survey on Product Derivation Methodsin Software Product Line. ISSN: 2458-6765. Mediterranean Telecommunications Journal Vol. 8, N° 1, January 2018.

[7] Rick Rabiser, Pádraig O‟Leary,* and Ita Richardson. Key Activities for Product Derivation in Software Product Lines. Journal of Systems and Software, 2011 - Elsevier

[8] Lothar Hotz and Andreas G¨unter, Thorsten Krebs. A knowledge based product derivation process and some ideashow to integrate product development. 2003. Kogs. www.informatik.uni-hamburg.

[9] Xin Peng, Liwei Shen, Wenyun Zhao. Feathers Implementation modeling based product derivation in software product line. International conference on software reuses ICRS 2008:High confidence software reuses in large systems. PP 142-53.

[10] Hugo Arboleda1,2 , Andres Romero1 , Rubby Casallas1 , and Jean-Claude Royer2. Product Derivation in a Model-Driven Software Product Line using Decision Models?

[11] Colin Atkinson, Joachim Bayer, and Dirk Muthig. Component-Based Product Line Development: The KobrA Approach. Fraunhofer Institute for Experimental Software Engineering (IESE), Sauerwiesen 6, D-67661 Kaiserslautern, Germany,

[12] PuLSE: A Methodology to Develop Software Product Lines Joachim Bayer, Oliver Flege, Peter Knauber, Roland Laqua, Dirk Muthig, Klaus Schmid, Tanya Widen, Jean-Marc DeBaud

[13] Soo Dong Kim, Hyun Gi Min, Jin Sun Her, Soo Ho Chang. DREAM: A Practical Product Line Engineering using Model Driven Architecture

[14] Marco Sinnema, Sybren Deelstra, Jos Nijhuis, Jan Bosch. COVAMOF: A Framework for Modeling Variability in Soware Product Families.2004. Spronger.

[15] Padraig O leary, Eduado Santanade Almedia Ita Richardson.The Pro-PD Process Model for Product Derivation within software product lines. 2012. Sciencedirect.

[16] A user-centered approach to product configuration in software product line engineering
R Rabiser - 2009 - ase.jku.at

[17] A formal approach to derivation of concurrent implementations in software product lines S. Yovine, I. Assayad, F.-X. Defaut, M. Zanconi, A. Basu. Unite Mixte de Recherche 5104 CNRS - INPG – UJF.2008.

[18] Pádraig O"Leary, Fergal McCaffery, Steffen Thiel, Ita Richardson. An Agile Process Model for Product Derivation in Software Product Line Engineering.

[19] Daren Nestor, Luke O'Malley, Patrick Healy, Aaron Quigley, Steffen Thiel. Visualisation Techniques to Support Derivation Tasks in Software Product Line Development. Proceedings of the 2007 …, 2007 - dl.acm.org

[20] Sybren Deelstra, Marco Sinnema, Jan Bosch. A Product Derivation Framework for Software Product Families

[21] Assessment for quality product derivation from a software product line reference architecture. F Losavio, O Ordaz, H Márquez - Revista Antioqueña de las …, 2015 - search.ebscohost.com

[22] Quality-based Heuristic for Optimal Product Derivation in Software Product Lines Quality-based Optimal Product Derivation in SPL Francisca Losavio, Oscar Ordaz. IEEE 2015.

[23] Nesrine Lahiani* , Djamal Bennouar†. A DSL-based Approach to Product Derivation for Software Product Line. Acta Informatica Pragensia, 2016, 5(2): 138–143 DOI: 10.18267/j.aip.90 Peer-reviewed paper.

[24] Gharibi, G., & Zheng, Y. (2016). ArchFeature: Integrating features into product line architecture. In Proceedings of the 31st Annual ACM Symposium on Applied Computing (pp. 1302-1308). New York: ACM. doi: 10.1145/2851613.2851764

[25] Tawhid, R., & Petriu, D. C. (2011). Product model derivation by model transformation in software product lines. In Proceedings of the 14th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (pp. 72-79). New York: IEEE. doi: 10.1109/ISORCW.2011.18

[26] Hector A. Duran-Limon, Carlos A. Garcia-Rios, Francisco E. Castillo-Barrera Rafael Capilla. An ontology based product architecture derivation approach IEEE

[27] J.M. Hunt. Organizing the asset base for product derivation IEEE. **INSPEC Accession Number:** 9089500. **DOI:** 10.1109/SPLINE.2006.1691578. **ISBN:**0-7695-2599-7

[28] A. Fantechi1 , S. Gnesi2 , G. Lami2 , and E. Nesti1. A Methodology for the Derivation and Verification of Use Cases for Product Lines. Springer 2004.

[29] FieSta: An approach for Fine-Grained Scope Definition, Configuration and Derivation of Model-Driven Software Product Lines Hugo Arboleda. 2009

[30] Defining and Validating a Multimodel Approach for Product Architecture Derivation and Improvement Javier González-Huerta, Emilio Insfran, Silvia Abrahão.