Journal of Theoretical and Applied Information Technology
31st December 2016. Vol.94. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

GOAL-BASED MODELING FOR REQUIREMENT TRACEABILITY OF SOFTWARE PRODUCT LINE

[1]ASAD ABBAS, [2]ISMA FARAH SIDDIQUI, [3*]SCOTT UK-JIN LEE

[1,2,3]Department of Computer Science and Engineering, Hanyang University ERICA, Ansan, South Korea
*[3]Corresponding Author
[1]asadabbas@hanyang.ac.kr, [2]isma2012@hanyang.ac.kr, [3]scottlee@hanyang.ac.kr

ABSTRACT

Software Product Line (SPL) is extensively used in industry for quick development with reusability of resources from domain engineering to application engineering. For testing the products from domain engineering to application engineering traceability of requirements are important. In sequential product development, it is easy to create the links between software artifacts. However, in SPL traceability links are difficult to create where multiple products from same domain with some variation point according to stakeholder. This paper proposes framework for traceability links in SPL processes i.e. domain engineering to application engineering artifacts by using goal base modeling. First step is to identify the variation points from domain feature model and trace the link at implementation level of SPL platform. Second step is to trace the links from each artifact of domain to application engineering for the development of final products. We have applied our approach on general SPL feature model and get the results of final products with zero constraint violation.
Keywords: *Software Product Line, Feature Model, Requirement Traceability.*

1. INTRODUCTION

Software Product Line (SPL) is an approach to develop the products with reusability of resources and development with less cost, time to market and increase the productivity. SPL consists two main engineering processes: first, Domain Engineering (DE) defines the complete scope of SPL with common and variable features that can be part of different products. Second, Application Engineering (AE) comprises of specific requirements from end user and develop the product by adopting resources from DE. Features from DE can be reusable in different products during AE with some variation points of each product [1-3]. Feature model is used to manage the common and variable features with cardinality relationship such as mandatory, alternative, optional and OR group [4-5]. Feature Model is a tree structure with parent and leaf nodes. The parent nodes are the compacted representation of functionality, however, leaf node represents the functionality of each feature in context of end user requirements [6]. Requirement traceability between artifacts of the development improves the consistency and help out to maintain the future developments and changes.

Requirement traceability is important for the maintenance in future testing and impact analysis of products. Moreover, it enhances the reusability of features and functionalities for further development [7]. Testing and maintenance of single product is easy to manage through requirement traceability because it requires only single product artifacts without considering variation points of the other projects. However, in multiple products, that shares common resources from DE to AE is difficult task because every product with different features require different traceability links and related information [8]. AE artifacts correspond to DE artifacts because of reusability of common and variable resources. Requirement traceability of SPL has two main process: 1) well managed variability in each product with variation points between every product, 2) binding of variability and common features in AE should be managed according to end user requirements (prioritization of requirement) [9]. Development of multiple products from SPL feature model fails due to improper requirement elicitations and week traceability links between artifact of DE and AE. Furthermore, the requirement traceability between variable features and variation points of each product is a complex task because of complex bindings and relationship of features. New variation seeds of features in

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

feature model according to change the requirements is cause of delay, high cost and low productivity.

In this paper we have proposed a method to present the requirements traceability from DE artifacts to AE artifacts in a systematic way. Proposed approach is a goal-based modeling where features of SPL from DE can be priorities according to end user requirements. Furthermore, the seeds according to end user requirements are mapped in goal base modeling to add new rationales in feature model. Features in DE have different to functionality priorities because of variation points in each product. Traceability link of DE artifacts to AE artifacts is based on the traceability relationships of each product. This relationship can be created on the basis of common goals of each product as SPL shares common features. Goal-Oriented requirement engineering is an appropriate method for requirement elicitation which automatically generates traceability links in each artifact of software development. Goal-base modeling is comprehensive method for traceability of goal for each requirement from different stakeholders, preference of goals from multiple stakeholders, core and optional features modeling and domain assumption representation.

The contribution of our work is to create the requirement traceability links between artifacts of SPL processes such as DE and AE. Our aim to find the final product configurations of SPL without any constraints violation with goal-base requirement traceability links.

The rest of paper is organized as: section 2 is comprises of feature model background, section 3 is related to work, section 4 is based on requirement traceability in sequential software development life cycle, section 5 SPL goal-base modeling, section 6 requirement traceability of SPL processes and finally section 7 is conclusion.

2. Feature Model Background

Leaf nodes of feature model present the functionality for developers and stakeholders. Kyo C. Kang proposed Feature-Oriented Domain Analysis (FODA) in 1990 for development of family of products that shares common resources for all products and differentiate with variability of features. The main objective of FODA is to represent the family of products according to common and variable features of SPL domain [10]. Feature-Oriented Reuse Method (FORM) is proposed by Kyo C. Kang in 1998 for reusability of features from domain repository of SPL. FORM is a module base approach i.e. remove the dependency between features and develop the domain engineering with independent modules to enhance the reusability of features in different products [11]. Fig. 1 depicts the simple example of mobile feature model [12].

Fig. 1 (a) depicts the complete scope of mobile phone SPL feature model. Fig. 1 (b) shows the different relationships between features. The feature orientation approach performed four main process as given below [13,14].

a) Domain Analysis: This process defines all resources (common and variable) and relationships (include and exclude) of core assets presented by feature model.
b) Domain Implementation: Implementation is performed in this process for all common and variable features for products derivation.
c) Requirement Analysis: Requirements from stakeholders are analyzed according to the scope of feature model and select suitable features for further product derivation according to end user requirements.

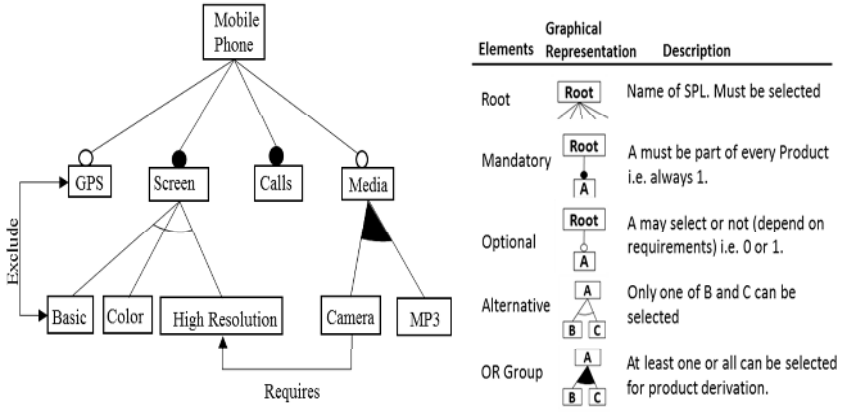

Figure 1. Mobile Phone Feature Model and Relationships

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

d) Product Derivation: All resources are already existing with proper functionality and also have the requirements according to end user. Combine all features with respect to relationships and constraints between them and final product derivation.

3. RELATED WORK

Yguarata et. al. proposed method for variants traceability of SPL feature model. UML notations are used to define the relationship between features and define the constraints at meta-model level. Meta-model is based on core assets of SPL and organized by UML model. Requirement traceability in each product with variants are need to be linked because common features are necessary to be the part of every product. Meta-model variant traceability approach is enable to trace the variation of products and enhance the testability of all products in one SPL scope by tracking the variation points [15].

Nicolas et. al. proposed the model-driven based AMPLET Traceability Framework (ATF) of SPL development process artifacts. ATF traceability management framework enhances the development in way to modify, import and export of features, query to find the variations of products and visualization of links between artifacts. Mata-model implementation such as DE of SPL, information of traceability links stored in data base repository that can be access by query to find the variation points [16].

Pourya et. al. presented Feature-Oriented Formal Language (FORMAL) for modeling the requirements of SPL. FORMAL modeling enables to model the existing and new requirements if added by end user. It supports the modularity of features to remove the dependency between features to increase the reusability. Decompose the features into unite modules to independent small features and enhance the reusability in different product derivations of SPL. Goals of FORMAL modeling are feature modularity, ease of development, modeling differences, features interaction modeling associative and commutative composition and precision [17].

Vikas et. al. proposed the method for requirement engineering with goal-oriented approach of SPL. Early stage of requirement engineering is critical because of understanding with requirements of stakeholder particularly when multiple stakeholders are involved in project. Author proposed Goal-Orient Requirement Engineering (GORE) for requirement elicitation from end user. Proposed method is called Comprehensive Modeling Language (CReML). Core and optional goals are important to priorities the requirements according to environmental features selection [9].

Dhungana et. al. proposed DOPLER approach to customize requirements of SPL domain. Customization is based on variability modeling of feature model. DOPLER is used to priorities the requirements of variability features, as common features are necessary in every product [18].

Lukas et. al. proposed method to incremental traceability of requirements of SPL. Variability modeling create variation in each product of SPL but always have reusability. Therefore, variability feature presents the same requirements but in different scenario with other related functionalities. Traceability links between artifacts of product development are based on same functionality descriptions [19].

Aforementioned studies are limited to traceability of requirements without considering complete development of SPL process i.e. DE to AE product derivations. Moreover, if requirements are changed and new features are required to add at the level of DE, it effects on complete SPL product family due to the inclusion of new variations and exclusion of some features. Therefore, traceability links and related information needs to change with respect to relationships and constraints between features. This study proposes the goal-base modeling for requirement traceability of complete development life cycle process of SPL i.e. domain engineering to application engineering, also handle traceability of new seeds (include new features) according to rationales of each feature.

4. REQUIREMENT TRACEABILITY IN SEQUENTIAL LIFE CYCLE:

Multiple tools and methods are used to traceability requirements and links. Caliber requirement management consist on repository of requirements that provides entire information of project with respect to requirements. Caliber enables the end-to-end requirements, impact analysis of requirements, traceability, priorities requirements, flexibility, integration of life cycle and security. Furthermore, Caliber enables to login from single account on multiple servers from different location to map the requirements from end user on single project [20].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Rational RequisitePro enable the features to navigate the trace links, maintaining, establishing and trace information in trace slice or matrices formats of project. The environment of project that enables the semi-automated method to retrieve information to generate dynamically trace links [21].

IBM has generated tool DOOR for requirement management. Requirements are stored in database in form of objects with hierarchal structure. Hierarchal structure allows the individual project versioning and enable to retrieve traceability link information [22].

Table 1. Analysis Of Requirement Traceability Of Existing Tools.

Traceability	Caliber-RM	Rational RequisitePro	DOORS
Links Management	Manual Requirements Development life cycle	Manual development life cycle	Manual and import development life cycle
Views	Traceability reports, diagrams and matrix	Traceability matrix and tree	Traceability tree and matrix
Queries	Traceability Links and requirements	Query identifies the functionalities on requirement attributes	Complete information including links and specific requirements
Extensibility	Not supported	Not supported	Create new links if change the requirements
Reusable Requirements	No	Yes	Yes
Catalog Evolution	No	Yes	No
SPL (DE and AE)	Not supported	Not supported	Not supported

Table 1 depicts the analysis of requirement traceability of existing tools about functionalities and limitations. Three requirement traceability tools has compared Caliber-RM, Rational RequisitePro and DOORS, traceability of SPL processes (DE and AE) are not supported by any of given tools.

5. SPL GOAL-BASE MODELING

SPL product derivation change the artifacts of development when requirement priorities has changed. Goals represent the stakeholder requirements that needs to achieve from system. System characterize the goals into primary and secondary goals. Stakeholder needs to fix on single point with project manager. Rationales needs to be clear from stakeholder that why the given goals are so important and need to achieve on priority basis. These rationales are linked with various responsibilities that required by end user, what system needs to do to achieve these goals. [9]. Goal based modeling successfully enables the reason and capture the requirements in different forms that can be functional, non-functional or any other agreements between holder and organization [23].

Fig. 2 describes the complete process for SPL goal modeling with new requirements and common functions that are used in every product. At initial stage of development goal model is structured for end user understanding, add new entities according to end user requirements. Identify the similar goals (functional and non-functional) in each SPL product, priorities each goals and find primary system requirements. Control loop handle the variability in product development if new variables are added in SPL DE. Finally, goal model is constructed that completely specifies the end user requirements.

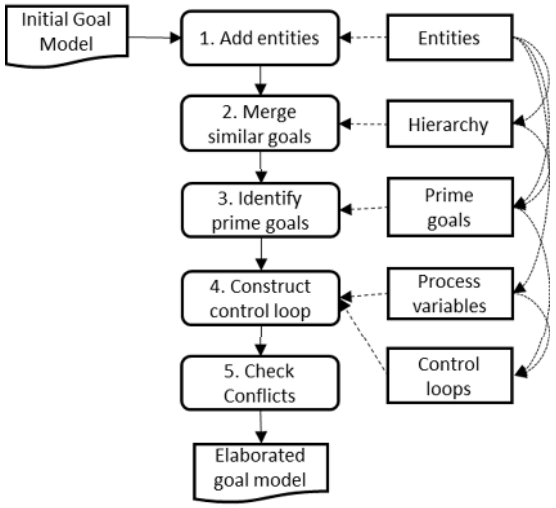

Figure 1. Goal Modeling of SPL Commonalities and Variability

6. REQUIREMENT TRACEABILITY OF SPL PROCESSES

Traceability of requirements from DE to AE artifacts, it is important to model the goals and strategy with clear primary and secondary requirements from stakeholders. If new goals introduced from stakeholders the rationales of system will be changed. Multiple stakeholders have different viewpoints as SPL is family of products and each product is varied on variation points and objectives. Developers need to trace the links of each variability point with respect to constraints.

Journal of Theoretical and Applied Information Technology
31st December 2016. Vol.94. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 2. Dimensions Of Traceability With Respect To SPL Processes.

Dimensions	Processes of SPL Engineering
Similarity	Both in Domain Engineering or Application Engineering
Variability Identification	Only in Application Engineering
Reuse of Variability	Application to Domain Engineering
Refinement	Both in Domain Engineering or Application Engineering
Versioning	Both in Domain Engineering or Application Engineerin

6.1. Spl Traceability Dimensions

We have considered five dimensions for traceability of SPL artifacts: Similarity of products (common features), variability realization (identify the variation points), variability reuse (enhance the variability with respect to relationships and constraints), refinement (new seeds add) and versioning (derivation of new products in scope of SPL). Table 2. Shows the traceability dimensions for SPL processes.

In Table 2 traceability dimensions has discussed for SPL DE and AE. Similarity identifies the common goals in complete SPL family and reuse as it is in all products from DE to AE. Variability identification point out all variation points (alternative, optional and OR group features) to differentiate products. Identification of variation points make easier to reuse these variability features in different products according to end user requirements. If end user requirements are changed during development of DE and AE, refinement is required in both SPL processes. At last versioning of products has developed when complete and precise requirements have received from end user.

6.2. Spl Feature Model New Goals:

New goals and variabilities are introduced in SPL product derivation and rationales of these goals also needs to change. Developers get the viewpoints and objectives of complete system from end user to find the commonalities and variabilities of SPL. Commonalities (mandatory) and variability (alternatives, optional, OR groups) are the features property that constructed at DE level. Identify the constraints (include and exclude) relationships between features and limitations of the system that is required to develop the AE. Stakeholder is involved in this complete process. Fig. 3 shows the complete framework of requirement traceability links of SPL process.

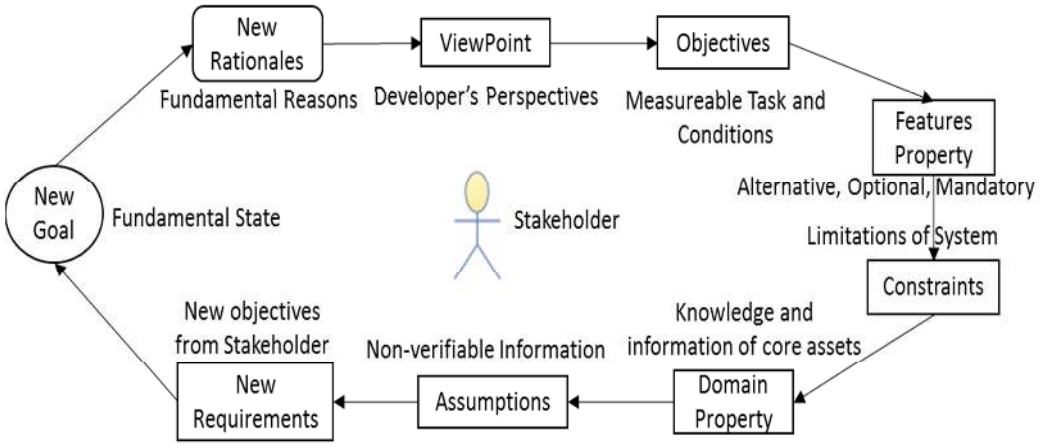

Figure 3. New Seeds Traceability Framework

Journal of Theoretical and Applied Information Technology
31st December 2016. Vol.94. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

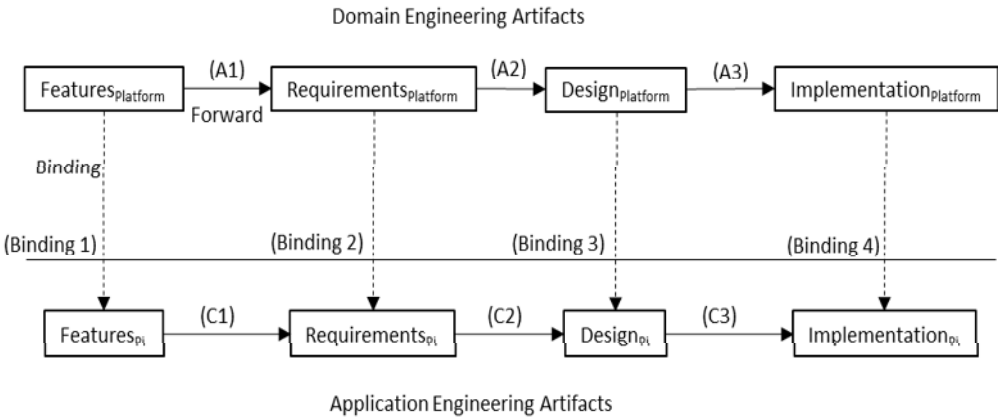

Figure 4. Traceability between SPL Artifacts

Fig. 4 describes the complete model of requirment traceability of DE and AE artifacts. DE artifact are consisted of features platform (core assets of SPL) transform to requirements, design and implementation of all features for reusability. In AE feature platfrom, specific SPL domain features are binding into product derivation, selection of features are based on end user requirements to develop specific ith product (pi) feature model. Tracebility links are developed between DE requirements and AE requirements and construct the design of pi. Implementation of features in pi is the reusability of already implemented features in domain (platform) of SPL.

Fig. 5 is a general feature model that have one common feature (F4), three variation points (vpf1, vpf2 and vpf3) and every leaf node have specific goals with relationships between each goal (alternative, optional and OR group).

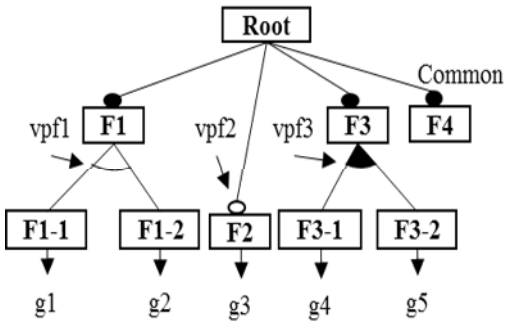

Figure 2. Goal-based feature model

Fig. 5 shows the platform of SPL with three variation points vpf1, vpf2 and vpf3, where each leaf node has goals that can be selected according to end user requirements. Multiple products can be derived from given feature model with vpf1 (variation point at F1), vpf2, vpf3 and F4. F4 is a common feature that must be part of each product therefore, we consider only variation points for traceability of goals in each product. Two constraints are given at vpf1 and vpf3 (mandatory OR group) in Fig. 5 as given below.

- Product 1: [vpf1\F1], where g1 excludes g2.
- Product 2: [vpf1\F1], where g2 excludes g1.
- Product 3: [vpf1\g1, vpf3\g4\g5], g1 exclude g2, g4 and g5 (one-to-many relationship).
- Product 5: [vpf1\g2, vpf3\g5\g4], g1 exclude g2, g4 and g5 (one-to-many relationship).
- Product 6: [vpf1\g2, vpf2\null, vpf3\g5\g4], g1 exclude g2, vpf2 (optional), g4 and g5 (one-to-many relationship).
- Product 7: [vpf1\g1, vpf2\null, vpf3\g5\g4], g1 exclude g2, vpf2 (optional), g4 and g5 (one-to-many relationship).

Table 3 shows all configuration of feature model in Fig. 5 that are defined by requirement traceability with given goals corresponding to five traceability dimensions given in Table 1. In all products 1 is presented selected features and 0 presented non-selected features

.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 3. SPL configurations by traceability of variation points

Software Product Line		SPL configurations goal traceability				
Variable features		F1-1	F1-2	F2	F3-1	F3-2
Products Binding of Features	P1	1	0	0	0	1
	P2	0	1	0	1	0
	P3	1	0	0	1	1
	P4	1	0	1	0	1
	P5	0	1	1	1	0
	P6	1	0	1	1	1
	P7	1	0	0	0	1
	P8	0	1	0	1	0
	P9	1	0	0	1	1
	P10	1	0	1	0	1
	P11	0	1	1	1	0
	P12	1	0	1	1	1
SPL Design	DE	vpf1, vpf2, vpf3, F2, F4(common and always 1)				
	Variation points	vpf1: F1-1, F1-2 (alternative) vpf2: F2 (optional) vpf3: F3-1, F3-2 (OR group)				

We have found with goal base requirement traceability; no constraint violation occurs in final product derivations of SPL during feature selections. Traceability of variation points and relationships between features are important for correct product derivations.

7. CONCLUSION

Requirement traceability is important to make and track the future changes in software development. Traceability of the requirements is an easy task in a single product, however in SPL, it is difficult to create traceability links in SPL artifacts because of change in requirements and variation points in each product. In this paper we have proposed a goal-based model for requirement traceability in SPL. Proposed framework provides flexibility to consider the multiple stakeholders and requirements changing in application engineering. Traceability link is from domain engineering to application engineering artifacts with new goals, rationales, objectives and constraint relationships in features of SPL. With our proposed approach, selected and non-selected features for final product derivations has identified with zero constraint violations.

REFERENCES

[1] K. Lee, K. C. Kang, and J. Lee, "Concepts and guidelines of feature modeling for product line software engineering," in *International Conference on Software Reuse*, 2002, pp. 62-77.

[2] J. Lee and K. C. Kang, "Feature binding analysis for product line component development," in *International Workshop on Software Product-Family Engineering*, 2003, pp. 250-260.

[3] A. Abbas, Z. Wu, I. F. Siddiqui, and S. U.-J. Lee, "An Approach for Optimized Feature Selection in Software Product Lines using Union-Find and Genetic Algorithms," *Indian Journal of Science and Technology,* vol. 9, 2016.

[4] Lytra, H. Eichelberger, H. Tran, G. Leyh, K. Schmid, and U. Zdun, "On the interdependence and integration of variability and architectural decisions," in *Proceedings of the Eighth International Workshop on Variability Modelling of Software-Intensive Systems*, 2014, p. 19.

[5] S. U.-J. Lee, "An Effective Methodology with Automated Product Configuration for Software Product Line Development," *Mathematical Problems in Engineering,* vol. 2015, 2015.

[6] A. Abbas, I. F. Siddiqui, and S. U.-J. Lee, "Multi-Objective Optimization of Feature Model in Software Product Line: Perspectives and Challenges," *Indian Journal of Science and Technology,* vol. 9, 2016.

[7] J. Cleland-Huang, O. C. Gotel, J. Huffman Hayes, P. Mäder, and A. Zisman, "Software traceability: trends and future directions," in *Proceedings of the on Future of Software Engineering*, 2014, pp. 55-69.

[8] I. Cabral, M. B. Cohen, and G. Rothermel, "Improving the testing and testability of software product lines," in *International Conference on Software Product Lines*, 2010, pp. 241-255.

[9] V. Shukla and G. Auriol, "Reinventing Goal-Based Requirements Modeling," in *CSDM (Posters)*, 2013, pp. 13-24.

[10] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson, "Feature-

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

oriented domain analysis (FODA) feasibility study," DTIC Document1990.

[11] K. C. Kang, S. Kim, J. Lee, K. Kim, E. Shin, and M. Huh, "FORM: A feature-; oriented reuse method with domain-; specific reference architectures," *Annals of Software Engineering,* vol. 5, pp. 143-168, 1998.

[12] C. Henard, M. Papadakis, G. Perrouin, J. Klein, and Y. Le Traon, "Assessing software product line testing via model-based mutation: An application to similarity testing," in *Software Testing, Verification and Validation Workshops (ICSTW), 2013 IEEE Sixth International Conference on*, 2013, pp. 188-197.

[13] K. C. Kang, J. Lee, and P. Donohoe, "Feature-oriented product line engineering," *IEEE software,* vol. 19, p. 58, 2002.

[14] F. Linden, K. Schmid, and E. Rommes, "The Product Line Engineering Approach," *Software Product Lines in Action,* pp. 3-20, 2007.

[15] Y. C. Cavalcanti, I. do Carmo Machado, P. A. da Mota, S. Neto, L. L. Lobato, E. S. de Almeida*, et al.*, "Towards metamodel support for variability and traceability in software product lines," in *Proceedings of the 5th Workshop on Variability Modeling of Software-Intensive Systems*, 2011, pp. 49-57.

[16] N. Anquetil, U. Kulesza, R. Mitschke, A. Moreira, J.-C. Royer, A. Rummler*, et al.*, "A model-driven traceability framework for software product lines," *Software & Systems Modeling,* vol. 9, pp. 427-451, 2010.

[17] P. Shaker, J. M. Atlee, and S. Wang, "A feature-oriented requirements modelling language," in *2012 20th IEEE International Requirements Engineering Conference (RE)*, 2012, pp. 151-160.

[18] Dhungana, D., Grünbacher, P., & Rabiser, R. (2011). The DOPLER meta-tool for decision-oriented variability modeling: a multiple case study. *Automated Software Engineering*, *18*(1), 2011, pp. 77-114.

[19] L. Linsbauer, S. Fischer, R. E. Lopez-Herrejon, and A. Egyed, "Using traceability for incremental construction and evolution of software product portfolios," in *Proceedings of the 8th International Symposium on Software and Systems Traceability*, 2015, pp. 57-60.

[20] D. M. Alghazzawi, S. T. Siddiqui, M. U. Bokhari, and H. S. A. Hamatta, "Selecting Appropriate Requirements Management Tool for Developing Secure Enterprises Software," *International Journal of Information Technology and Computer Science (IJITCS),* vol. 6, p. 49, 2014.

[21] P. Mäder, P. L. Jones, Y. Zhang, and J. Cleland-Huang, "Strategic traceability for safety-critical projects," *IEEE software,* vol. 30, pp. 58-66, 2013.

[22] G. Beier, A. Figge, R. Müller, U. Rothenburg, and R. Stark, "Supporting Product Development through Cross-Discipline Dependency-Modeling–Novel Approaches for Traceability-Usage," *Lecture Notes on Information Theory Vol,* vol. 1, 2013.

[23] A. Palmieri, P. Collet, and D. Amyot, "Handling Regulatory Goal Model Families as Software Product Lines," in *International Conference on Advanced Information Systems Engineering*, 2015, pp. 181-196.