Journal of Computer and Communications, 2014, 2, 101-108
Published Online January 2014 (http://www.scirp.org/journal/jcc)
http://dx.doi.org/10.4236/jcc.2014.22018

Scientific Research

Feature Modeling and Variability Modeling Syntactic Notation Comparison and Mapping

Wahyudianto, Eko K. Budiardjo, Elviawaty M. Zamzami

[1]Faculty of Computer Science, Universitas Indonesia, Depok, Indonesia; [2]Departement of Computer Science, Universitas Sumatra Utara, Medan, Indonesia.
Email: wahyudianto@ui.ac.id eko@cs.ui.ac.id; elvi_zamzami@usu.ac.id

Received November 2013

ABSTRACT

Feature Model (FM) became an important role in Software Product Line Engineering (SPLE) field. Many approaches have been introduced since the original FM came up with Feature Oriented Domain Analysis (FODA) introduced by Kang in 1990. The main purpose of FM is used for commonality and variability analysis in domain engineering, to optimize the reusable aspect of software features or components. Cardinality-based Feature Model (CBFM) is one extension of original FM, which integrates several notations of other extensions. In CBFM, feature model defined as hierarchy of feature, with each of feature has a cardinality. The other notation to express variability within SPLE is Orthogonal Variability Model (OVM). At the other hand, OMG as standard organization makes an effort to build standard generic language to express the commonality and variability in SPL field, by initiate Common Variability Language (CVL). This paper reports the comparison and mapping of FODA, CBFM and OVM to CVL where need to be explored first to define meta model mapping of these several approaches. Furthermore, the comparison and mapping of those approaches are discussed in term of R3ST (read as "REST") software feature model as the case study.

KEYWORDS

Comparison and Mapping; FODA; Cardinality-based Feature Model; Orthogonal Variability Language; Common Variability Language; Feature Model; R3ST Software

1. Introduction

Software Product Line Engineering (SPLE) was coming from Product Line (PL) field. Feature Model (FM) which firstly introduced by Kang with Feature Oriented Domain Analysis (FODA) became a core part of SPLE research and development [1]. The original FM has been extended by several approaches [2], one of them is Cardinality-based Feature Model (CBFM) [3]. Another approach beside used of FM notation to express variability is Orthogonal Variability Model (OVM) [4]. To conform this approach, Common Variability Langua-ge as new emerging standard was initiated by OMG to develop general language for expressing variability within SPL [5].

Our main goal in this research is to define the mapping of FODA, CBFM and OVM to CVL. We want to study the comparison and relation mapping of tree approaches to CVL. Is the FODA, CBFM, OVM and CVL has relation?

In this paper we report the recent progress of our research. We analyze the comparison of FODA, CBFM and OVM with CVL. We identify the mapping of tree approaches to CVL. And at the last, we describe the relation of FODA, CBFM and OVM with CVL.

To organize this paper we describe the related background in Section 2, the grounded theory of feature modeling in SPLE, FODA, CBFM, OVM and recent state of CVL. The related work of this research is discussed in Section 3. The comparison and mapping analysis are discussed in Section 4, and also relation of FODA, CBFM and OVM with CVL. Study case of this comparison and mapping using R3ST software prototype are explained in Section 5. And finally we conclude and discuss the feature work in Section 6.

2. Background

2.1. Feature Oriented Domain Analysis Feature Model

The Feature Model was came up with Feature Oriented Domain Analysis (FODA) concept, that introduced by

Kang at 1990 (original FM) [1]. FODA is concept that used to analyze the domain problem on SPL.

From sample FODA at **Figure 1**, is show that the tree is composed by feature name, mandatory feature, optional feature, alternative feature, and composition rule.

2.2. Cardinality-Based Feature Model

Cardinality-based Feature Model (CBFM) was integrated several original FODA extensions. It is hierarchical feature model, (see **Figure 2**) where each feature has cardinality. The notation of feature cardinality is [*m*..*n*], with interval from m to n, where $m \in Z \wedge n \in Z \cup \{*\} \wedge 0 \leq m \wedge (m \leq n \vee n = *)$. The other additional notation is group cardinality for each feature group, where feature can be arranged. A group cardinality has interval from m to n with notation [m-n], where $m, n \in Z \wedge 0 \leq m \leq n \leq k$, and where k is the number of features in the group [3].

2.3. Orthogonal Variability Model

Orthogonal Variability Model (OVM) is the other concept to express variability on SPL process beside FM. OVM express variability using Variation Point (VP), Variant (V) and Dependency. A VP notation is representing the subject of variability and V notation is representing the object of variability, and can be used to trace with other artifacts [4]. For example see **Figure 3**.

2.4. Common Variability Language

Several approaches to express the variability in SPLE are described above. They all have the same goal, but come with different approaches. Object Management Group (OMG) initiative to develop an approach that can be used generically, by introducing the Common Variability Language (CVL) as a standard for expressing variability. CVL as stated in the specification [5], is a language for the domain independent, used to define and solve the problems of variability in the Domain Specific Language (DSL). DSL is a combination of a domain expert competence for one or more product line. The concrete syntax

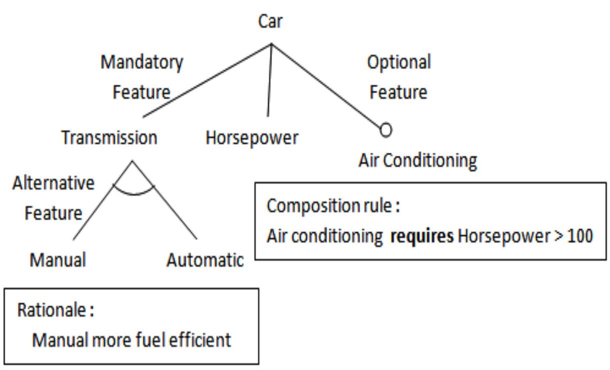

Figure 1. Sample FODA feature model [1].

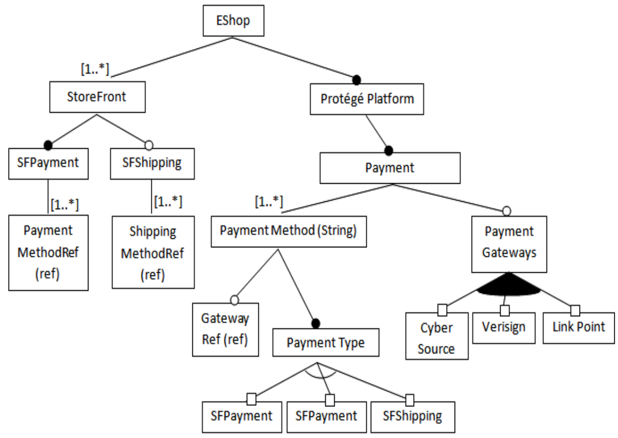

Figure 2. Sample Cardinality-based Feature Model Excerp [3].

of CVL is using Variability Specification (VSpec) Tree to model the feature [5]. The VSpec is a FM like using tree diagram as we can see in **Figure 4**.

3. Related Work

The technical report from SINTEF [6] describes the mapping between the components contained in the feature diagram and CVL element (see **Table 1**). This report is submitted on OMG Request for Proposal (RFP). This is what underlying the subsequent development of CVL.

Another initiative is a comparison of variability modeling of the existing approach [7]. The comparison based on Feature Modeling (FM) and Decision Modeling (DM). This research concluded that the FM can express both commonality and variability, whereas DM only express variability.

Reference [8], comparing characteristic and metamodel contained in the existing features language. This research proposed metamodel (**Figure 5**) that must exist to bring together all the features of existing models.

From unified feature metamodel mentioned at **Figure 5**, the common component that must be present is:

1) Feature; 2) Optional Feature; 3) Mandatory Feature; 4) Alternative Feature; 5) Feature Cardinality; 6) Feature Properties; 7) Exclude; 8) Requires; 9) Constraints; and 10) Label.

4. Syntactic Notation Comparison

The comparison of concrete syntax notation is based on literature study of four approaches. This research compared the 12 components of feature model and variability model and the result is shown in **Table 2**. The explanation of the comparison result is outlined below.

4.1. Feature

All approaches have feature notation except OVM. The

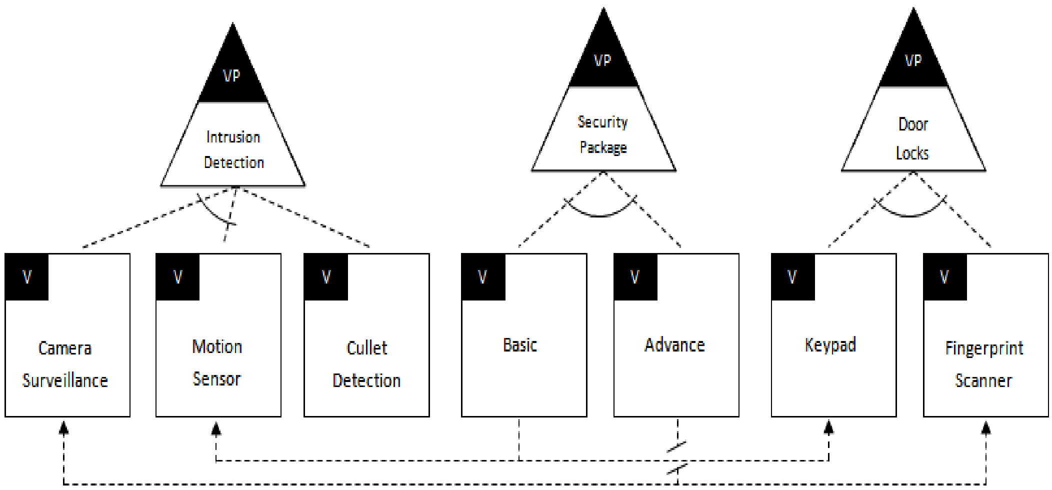

Figure 3. Sample orthogonal variability model [4].

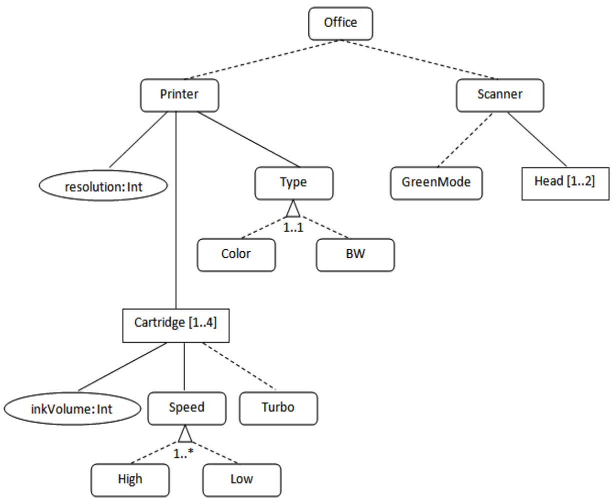

Figure 4. Sample vspec tree of CVL [5].

Table 1. Feature diagram and CVL mapping [6].

Semantics	Symbol	CVL Element
Mandatory		:CompVar
Optional		:Iterator Lower =0 Upper = 1
AND		:CompVar
OR		:Iterator Lower =1 Upper = -1 IsUnique = true
XOR		:Iterator Lower =1 Upper = 1
Multiplicity	[m..n]	:Iterator Lower = x Upper = y IsUnique = true

FODA directly use feature name as feature notation. The CBFM uses feature name inside a rectangle, and CVL uses rounded rectangle with its name inside.

4.2. Variable

Approaches that have variable notation are CBFM and CVL. The CBFM uses rectangle with attribute name followed by attribute type in a bracket inside. And the CVL notation uses an oval (ellipsis) containing variable name and its type separated by a colon.

4.3. Variation Point

Variation Point only expressed explicitly by OVM. The notation of VP is triangle consisting of black triangle at the top and name of variation points at the bottom. VP on CVL are bound directly to VSpec that have semantic, but not expressed explicitly in any normative concrete syntax. While other approaches express VP implicitly on feature tree only.

4.4. Variant

Variant only expressed explicitly by OVM with represented as rectangle with a black rectangle in the upper left corner and name of variant in the middle of it. While the others did not explicitly express this.

4.5. Mandatory

All approaches use solid edge to express mandatory fea-

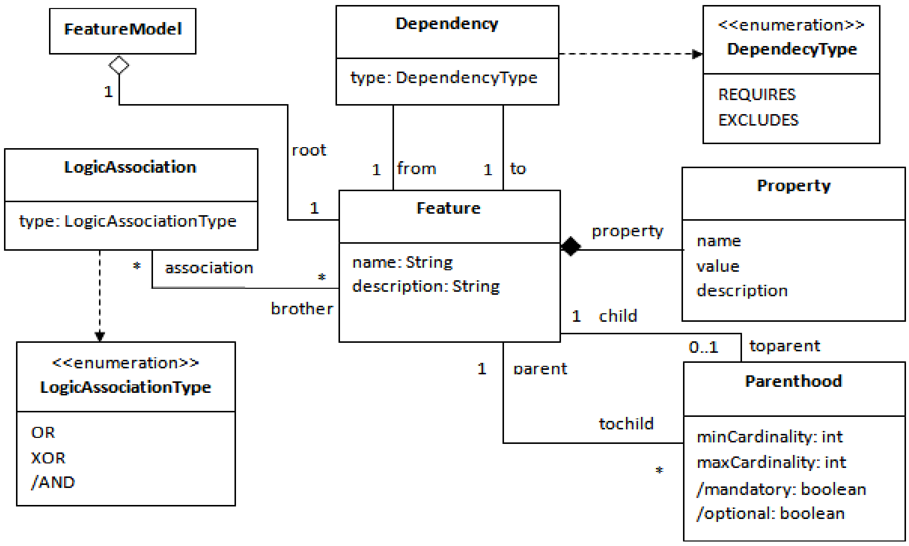

Figure 5. Unified feature metamodel [8].

Table 2. Syntactic notation of modeling mapping.

Component	FODA FM [1]	CBFM [3]	OVM [4]	CVL [5]
Feature	Car	StoreFront	X	PrinterPool
Variable	X	minutes(Int)	X	inkVolume: Int
Variation Point	X	X	VP [name]	X
Variant	X	X	V [name]	X
Mandatory				
Optional				
AND				
OR	X		[min..max]	1..*
XOR			[min..max]	1..1
Cardinality	X	[m..n],‹m..n›	min..max	0..n , n..m
Clone	X	[n..m] F	X	Printer [1..*]
Constraint	Composition rule: Air conditioning requires Horsepower > 100	context B: C implies D	requires_v_vp	fax implies printer

ture, but CBFM uses black circle at the lower end of it.

4.6. Optional

The optional feature for FODA and CBFM expressed with solid edge and white circle at the lower of it, while OVM and CVL used dashed edge.

4.7. AND

All approaches express the AND feature with included all solid edge. This means that the node under the parent which uses solid edge must be selected during configuration or must have in final product.

4.8. OR

OR feature expressed by CBFM, OVM and CVL, the FODA does not express this. CBFM uses black arc joining the solid edge. CBFM uses term OR-Group for this component. The OVM can express this, by give 1.* to group cardinality. CVL express or feature using term Group Multiplicity that uses small triangle with multiplicity 1.*. The FODA approach does not express OR feature.

4.9. XOR

XOR feature with the other term alternative feature is expressed by all approaches. FODA and CBFM use white arc joining solid edge. OVM uses white arc joining the dashed edge with default multiplicity 1..1. CVL uses small triangle with multiplicity 1..1.

4.10. Cardinality

The feature cardinality expressed by all approach except FODA. CBFM have [m..n] notation for Feature Cardinality and < m..n > notation for Group Cardinality. OVM uses [min..max] in square bracket notation and using term Multiplicity. CVL using term Multiplicity for this component and m..n notation for this component.

4.11. Clone

Feature Clone just expressed by CBFM and CVL. CBFM uses [m..n] in square bracket at above the feature notation. In CBFM [0..n] cardinality express Optional Cloneable Feature and [m..n] cardinality express Mandatory Cloneable Feature. In CVL, feature clone expressed using Variability Classifier (VClassifier) that uses rectangle with name followed by multiplicity notation in a square bracket inside.

4.12. Constraint

The feature constraint have expressed by all approaches with different form. FODA uses mutex and requires form in rectangle with title Compositional Rule. CBFM uses OCL and the best known are implies and excludes form that expressed in textual model. The OVM using requires and excludes form with dashed arrow that can be used for V and V, V and VP, and also VP and VP. And at last CVL uses parallelogram with basic constraint (subset of OCL) inside, even though in addition CVL allows using other constraint languages, including more complete OCL.

5. Comparison and Mapping Case Study

In this section we present the case study to compare four approaches that we have mapped and compared at section 4. The case study is feature model of R3ST (read as “REST”), the software that we still working on it. R3ST stands for Requirement Recovery and Reconstruction Software Tools. The main functionality of R3ST is to automate the reverse engineering that capture the End-to-End interaction of user and system until defined some goal of the system, and recover the Use Case model, and at the end, this tools provide SRS document of the reversed system [9]. For this case study, we focus on the End-to-End Interaction capture prototype, so we can clearly understand about comparison of four approaches to express commonality and variability.

FODA express the R3ST feature model with 4 elements as we can see at **Figure 6**. The name feature, mandatory feature, optional feature, and XOR Group feature notation. At here we still did not need the constraint notation. So we just ignore it.

The other approach is CBFM, express R3ST FM with different notation from FODA, as we can see at **Figure 7**. The main deference meaning is OR Group notation. With FODA we can’t express OR Group Feature. This OR Group means that we can use just one database engine or two or three. This can’t be expressed in FODA.

The OVM expresses the variability of R3ST is given in **Figure 8**. It is very different from FODA and CBFM, since this OVM is focusing on express variability not commonality. The OVM of R3ST consists of four variant point, the variability that OVM expressed.

The CVL model of R3ST is described in **Figure 9**. It is closely like CBFM since the CVL uses tree like notation to express the variability. The difference of both approaches is just the notation, but at the meaning level it is closely alike.

By R3ST case study given above, we can get the comparison of the four approaches. First, we can see that original FODA can’t express OR feature and cardinality, so we can just make the feature be an option for all children feature. Second, very different notation from others is shown by OVM because it’s focusing on expressing variability by having explicit variant and variant point notation. And the third, the CBFM and CVL have

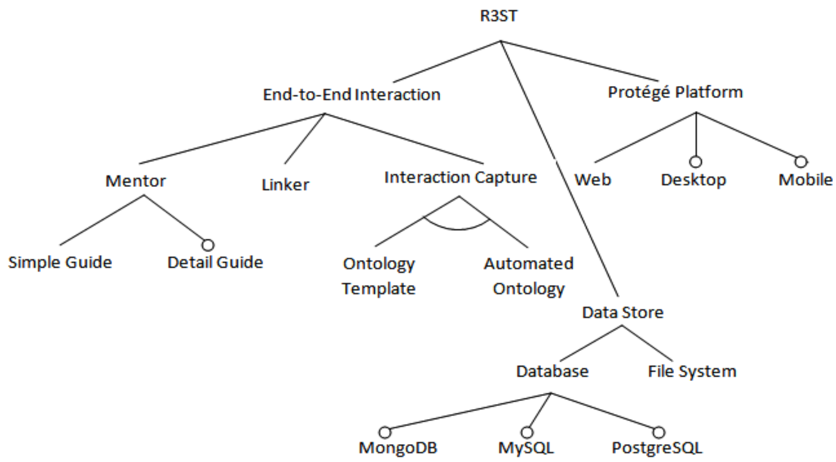

Figure 6. R3ST Feature model using FODA.

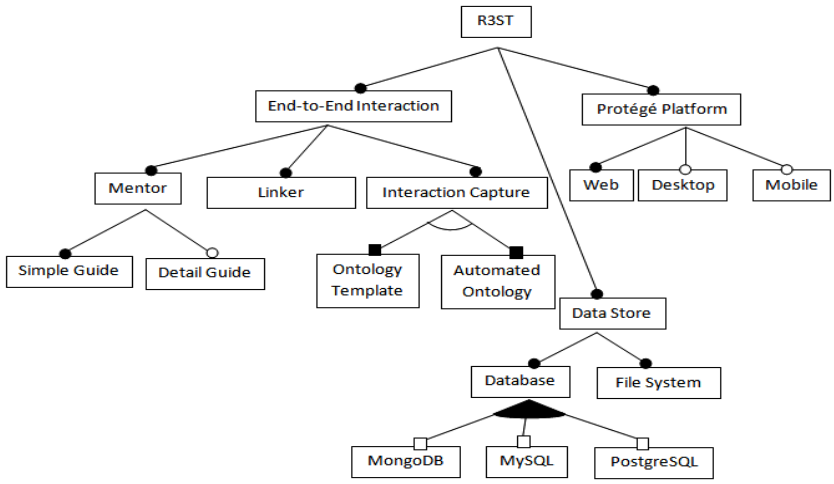

Figure 7. R3ST Feature model using CBFM.

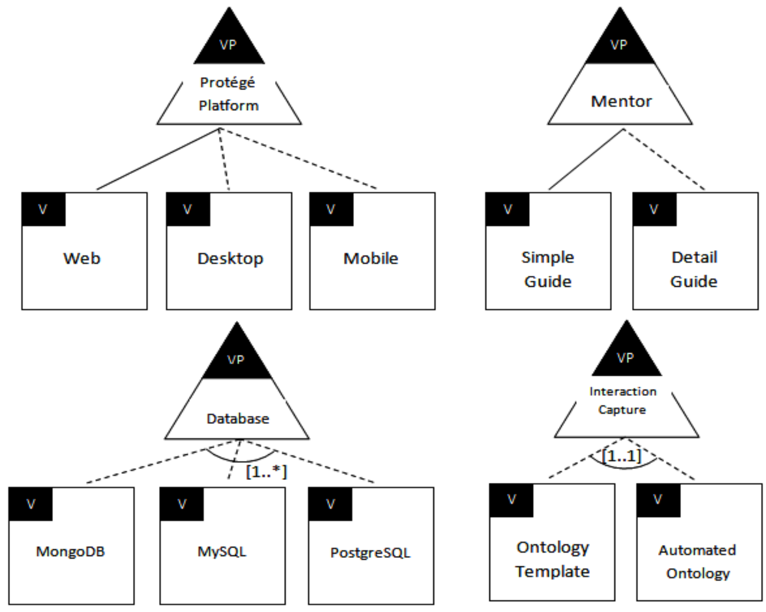

Figure 8. R3ST Variability model using OVM.

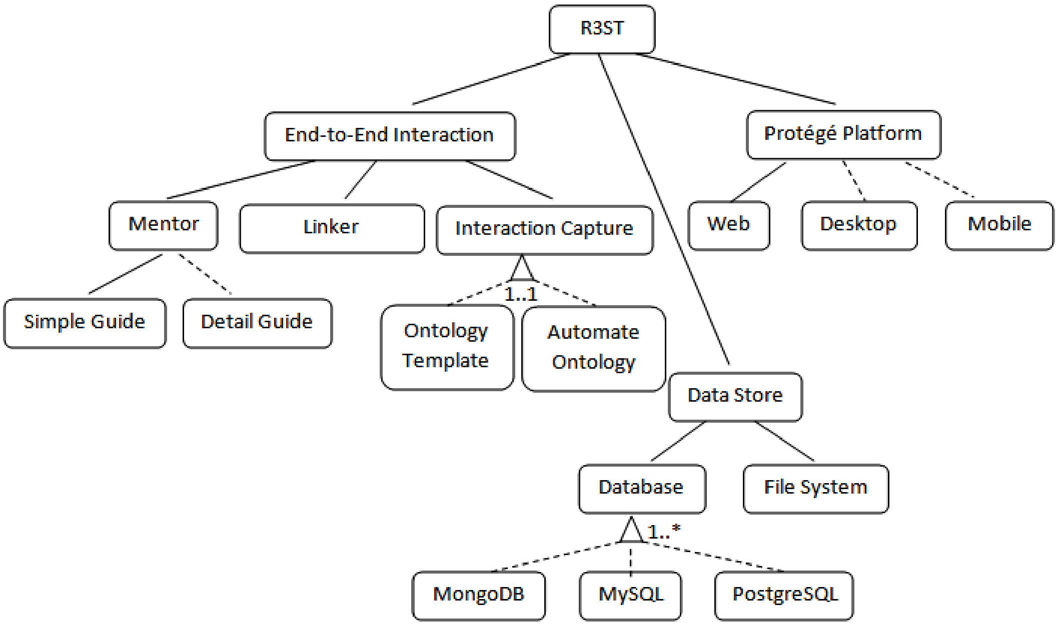

Figure 9. R3ST Feature model using CVL.

closely like notation, with the tree and cardinality of the feature.

6. Conclusion and Feature Work

6.1. Conclusion

In this paper, we compare the several approaches of feature modeling and variability modeling and do the mapping of their component. The original FODA as the first introduce the FM has limited notation to express commonality and variability in SPL, as already explained that just have notation for feature, mandatory, optional, AND, XOR, and constraints. The CBFM as an extension of original FODA is more expressive to describe commonality and variability in SPL with introducing cardinality concept in FM. The OVM as proposed to document variability is expressive and only focus to describe variability in SPL, but not the commonality. The other expressiveness of OVM is the traceability with other artifacts. And the last, CVL as proposed for specifying and resolving variability is more expressive to describe variability in SPL. This is in line with the objectives of the CVL as an independent domain and the language proposed by the OMG as a standard for modeling the variability in SPL.

6.2. Feature Work

For feature work, after we define this comparison and mapping, we plan to define the relational mapping on metamodel level, after the syntax notation mapping is done by this paper. Furthermore we plan to define the technique to integrate the several approaches with CVL as new standard for variability modeling in SPL.

REFERENCES

[1] K. C. Kang, S. Cohen, J. Hess, W. Nowak and S. Peterson, "Feature-Oriented Domain Analysis (FODA) Feasibility Study," Technical Report CMU/SEI-90-TR-021, Carnegie Mellon University Software Engineering, 1990.

[2] K. C. Kang, "FODA: Twenty Years of Perspective on Feature Modeling," *Proceeding of VaMoS*'10, Vol. 37 of ICB-Research Report, Universität Duisburg-Essen, 2010, p. 9.

[3] K. Czarnecki and P. Kim, "Cardinality-based Feature Modeling and Constraints: A Progress Report," Proceeding of International Workshop on Software Factories, 2005.

[4] K. Pohl, G. Bockle and F. Linden, "Software Product Line Engineering: Foundations, Principles and Techniques," Springer, 2005.

[5] Ø. Haugen, "Common Variability Language (CVL)," OMG Revised Submission, 2012. http://www.omgwiki.org/variability/lib/exe/fetch.php?id=start&cache=cache&media=cvl-revised-submission.pdf

[6] F. Fleurey, Ø. Haugen, B. Møller-Pedersen, G. K. Olsen, A. Svendsen and X. Zhang, "A Generic Language and Tool for Variability Modeling," Technical Report SINTEF A13505, SINTEF, Oslo, 2009.

[7] K. Czarnecki, P. Grünbacher, R. Rabiser, K. Schmid and A. Wasowski, "Cool Features and Tough Decisions: A Comparison of Variability Modeling Approaches," *Proceeding of VaMoS*'12, ACM, 2012.

[8] S. Sepúlveda, C. Cares and C. Cachero, "Towards a Unified Feature Metamodel: A Systematic Comparison of Feature Languages," *Proceeding of 7th Iberian Confe-*

rence on Information Systems and Technologies (*CISTI*), IEEE, 2012.

[9] E. M. Zamzami, E. K. Budiardjo and H. Suhartanto, "Requirements Recovery Using Ontology Model for Capturing End-to-End Interaction of Proven Application Software," *IJSEIA International Journal of Software Engineering and Its Applications*, SERSC, Vol. 7, No. 6, 2013, pp. 425-434.