Advanced Science and Technology Letters
Vol.49 (SoftTech 2014), pp.73-78
http://dx.doi.org/10.14257/astl.2014.49.16

Transformation of Cardinality-Based Feature Model to Common Variability Language Model

Wahyudianto[1] and Eko K. Budiardjo[2]

Faculty of Computer Science, University of Indonesia,
Depok, Indonesia
[1] wahyudianto@ui.ac.id, [2] eko@cs.ui.ac.id

Abstract. Cardinality-Based Feature Model (CBFM) is an approach for feature modeling, which integrates several extensions of Feature Oriented Domain Analysis (FODA) Feature Model (FM), where the main function of FM is to express commonality and variability in Software Product Line (SPL). CBFM was introduced by applying cardinality concept to feature model. Beside the CBFM, there are many other approaches that included in FM genealogy. As an international standard organization, Object Management Group (OMG) gives effort on developing Common Variability Language (CVL) as a standard notation to express commonality and variability in SPL. This paper explain the development of a transformation system from CBFM to CVL Model (CVLM) using model-to-model transformation approach within Model Driven Engineering context and aim to help automation process of CBFM to CVLM transformation, that maintain semantic meaning of transformed models.

Keywords: CBFM2CVLM, Model-to-Model Transformation, EMF QVTo, Feature Model, Cardinality-Based Feature Model, Common Variability Language, Model Driven Engineering.

1 Introduction

Feature Model (FM) is a concept that is most widely used in Software Product Line (SPL) [1]. FM was introduced along with the emergence of Feature Oriented Domain Analysis (FODA) concept, with the aim to model commonality and variability in SPL [2]. Over the time, FODA has been developed using several approaches, one of those approaches is Cardinality-Based Feature Model (CBFM). CBFM appears to propose the application of cardinality concept in FM, which has had a formal semantics [3]. One of several tools to graphically represent CBFM is Software Product Line Online Tools (SPLOT). SPLOT provide SXFM Spec, which has translated into an EMF Ecore Model [4].

With many approaches to express commonality and variability in SPL, Object Management Group (OMG) initiated to develop Common Variability Language (CVL) as a standard notation for domain-independent language [5]. CVL Model (CVLM) concept has been proposed by OMG and developed a prototype CVL2TOOL, but it has not determined when it will be launched [6].

ISSN: 2287-1233 ASTL

The main goal of this work is to develop a prototype for transformation from CBFM to CVLM. We want to study on how to develop a transformation system from CBFM to CVLM using Model Driven Engineering (MDE) approach. MDE approach is used in order to maintain the truth of semantic meaning, so that the original model has the same semantic meaning with the transformed model. This transformation is implemented through a mapping and transformation scheme in SPLOT SXFM Spec and CVL2TOOL. At the end of this research, transformation and mapping within CBFM and CVLM can provide the beneficial of using CVL as a standard which proposed by OMG to express commonality and variability in SPL.

To organized this paper we describe the related background in section 2, the briefly explain the concept of CBFM, CVL, and Model Transformation. In section 3, we explain the CBFM2CVLM Transformation. And finally we conclude and discuss the future work in section 4.

2 Theoretical Foundation

Cardinality-Based Feature Model (CBFM) is an approach that proposed the consolidation of cardinality concept within feature model concept, which has a formal semantics [1] and also has complete notation [7]. One of several tools to represent CBFM is Software Product Line Online Tools (SPLOT). SPLOT provide SXFM Spec as the basis of feature model, which has been translated into Eclipse Modeling Framework (EMF) Ecore Model [4].

Common Variability Language (CVL) is a standard notation developed by OMG, to express commonality and variability within SPL. The CVL model (CVLM) called by VSpec, which is a representation of the feature [5]. One of the CVL prototypes is CVL2TOOL [6], refer to the CVL specification of Revised Submission version [8].

Model Transformation is a concept to transform the model into desired output, that usually determines the type of model that can be accepted as input and the output according to the metamodel owned by the model [9]. Model-to-model transformation is the one type of model transformation that uses abstract syntax definition for Domain Specification Language (DSL). This transformation can be performed using Query/View/Transformation (QVT) Operational Mapping as a tool for transforming that provided QVTo by Eclipse Modeling Framework (EMF) [10].

3 CBFM2CVLM Transformation

3.1 Transformation Scheme

The transformation is performs by utilizing EMF in Eclipse environment, which is one of its functions, that can be used to develop a model-to-model transformation through Ecore metamodel. Since the transformation is perform based on the metamodel level, it is necessary to involve SPLOT SXFM Spec and CVL Ecore using

QVTo as a language for model-to-model transformation. Metamodel level utilization as the basis of this transformation is aimed to keep the semantics contained in the model remains the same, although the notations have changed, from SXFM model notation to CVL model notation. So the semantic correctness is need to be considered, and will be concerning on this transformation, using the syntax and metamodel mapping. Transformation scheme of CBFM2CVLM is illustrated in Fig 1.

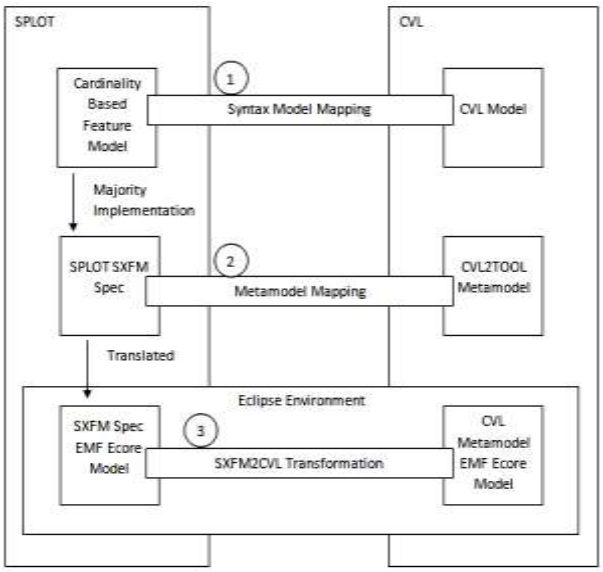

Fig. 1. CBFM2CVLM Transformation Scheme.

Using the illustration of transformation scheme in Fig. 1, thare are three stages that will be performs, there are syntax mapping, metamodel mapping, and SXFM transformation to CVL. Syntax mapping stage is our previews work that compares the several approah to expressing commonality and variability and perform mapping to each elements of thet approach [11]. The next section of this paper will explain the metamodel mapping that used in SXFM to CVL transformation as foundation.

3.2 Metamodel Mapping

Mapping SXFM and CVL is perform by mapping elements contained in each metamodel. Metamodel of SXFM and CVL obtained using EMF Ecore model from SPLOT-SXFM Spec (see Fig. 2) and CVL2TOOL prototype (see Fig. 3).

Based on the metamodel of SXFM and CVL, SXFM mapping to CVL metamodel can be performed. This mapping is done based on the elements contained in the syntax mapping that has been done [11]. Mapping results of SXFM metamodel to CVL metamodel is described on Table 1.

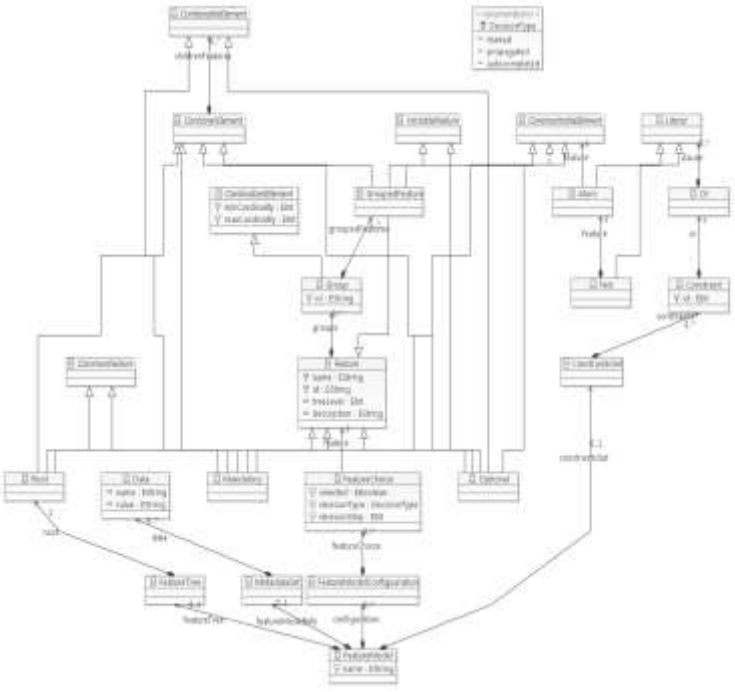

Fig. 2. SPLOT SXFM Spec [4].

In the Table 1, the tenth element is based syntax mapping on the previews work [11], which contained in both CBFM and CVLM. The *Feature* mapped to *VSpec* for feature element. Property included is *name*, while *id, treeLevel*, and *description* are not included in the mapping. Because *Feature* and *VSpec* is an abstract, so the mapping will be perform on an inheritance of that class. SXFM does not have a metamodel element for *variable feature*, so there is no mapping for *variable* element. *Mandatory* element is mapped to *Choice* by returning *true* to *isImpliedByParent*. *Optional* is also mapped to *Choice* but the *isImpliedByParent* property is returned *false*, although CVL returned it *false* by default. There is no metamodel element in SXFM for AND components, but can be arranged when at least two *child* of a feature is a *mandatory feature*. So the AND component is map *Mandatory* to *Choice*.

OR refers to the syntax mapping, is a *feature group* that has 1..* cardinality. So the mapping of this element, from *Group* by using *minCardinality* and *maxCardinality* as properties, mapped to *MultiplicityInterval* with *upper* and *lower* as properties. In EMF, the star (*) term for *cardinality* called *unbound multiplicity* with a value of -1. XOR is same as the *OR*, but *XOR* use 1..1 cardinality. For *Cardinality* element, the mapping is performing in *CardinalizedElement* to *MultiplicityInterval*. XSFM does not have a metamodel element for *feature clone*, so for Clone, mapping does not perform, though CVL has *VClassifier*. And for the last element is *Constraint*. *Constraint* from SXFM is mapped to *Constraint* of CVL.

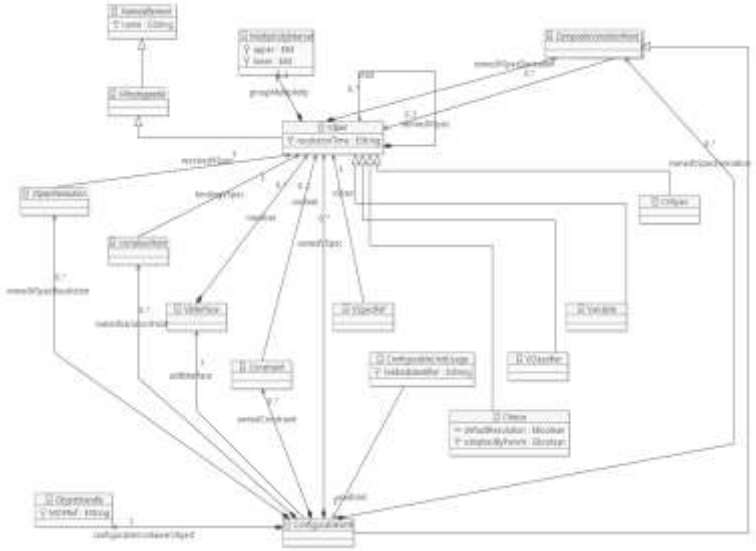

Fig. 3. CVL Metamodel Excerpt [7].

Table 1. SXFM to CVL Metamodel Mapping

Element	SXFM Spec	CVL Metamodel
Feature	Feature	VSpec
Variable	-	Variable
Mandatory	Mandatory	Choice
Optional	Optional	Choice
AND	Mandatory	Choice
OR	Group	MultiplicityInterval
XOR	Group	MultiplicityInterval
Cardinality	CardinalizedElement	MultiplicityInterval
Clone	-	VClassifier
Constraint	Constraint	Constraint

4 Conclusions

CBFM2CVLM scheme perform the transformation of feature model contained in SPLOT to CVL model. CBFM2CVLM transformation process is based on syntax notation and metamodel mapping. The metamodel mapping using SXFM Spec EMF Ecore Model and CVL Metamodel, there are some elements that do not have a relation mapping, which are variable feature and clone feature, because SXFM Spec does not define those two elements.

Based on CBFM2CVLM transformation scheme, syntax mapping, and metamodel mapping, this study aims to automate the process transformation of CBFM2CVLM of

SPLOT SXFM Spec and CVL2TOOL on Eclipse environment, that also maintaining the truth of semantic meaning of the transformed model. By this transformation, we can reuse the product families that have been modeled in SXFM, to be used on product family development using CVL as a new standard proposed by OMG to express commonality and variability.

The feature works is applying this CBFM2CVLM transformation scheme, using SPLOT and CVL integration model, so that can be embedded on R3ST (our research project on requirement recovery and comprehesion) , which can be the another way to software product line comprehension using the feature modeling concept.

The verification using model checking for model transformation can also be developing to the CBFM2CVLM transformation scheme. So in the end, it can show whether this transformation process is correct or not by model checking. And for the last, it needs more feature model that used as inputs in transformation process, as a proof of CBFM2CVLM transformation scheme perfection. With combining all these suggestions, it can be used as a way to prove that CBFM2CVLM transformation scheme is a sound and complete scheme.

References

1. Czarnecki, K., Grünbacher, P., Rabiser, R., Schmid, K., Wasowski, A.: Cool Features and Tough Decisions: A Comparison of Variability Modeling Approaches. In Proceeding of VaMoS'12, ACM (2012)
2. Kang, K. C., Cohen, S., Hess, J., Nowak, W., Peterson, S.: Feature-Oriented Domain Analysis (FODA) Feasibility Study. Technical Report CMU/SEI-90-TR-021. Carnegie Mellon University Software Engineering Institute (1990)
3. Czarnecki, K., Helsen, S., Eisenecker, U. W.: Formalizing cardinality-based feature models and their specialization. Software Process: Improvement and Practice, 10(1):7–29 (2005)
4. Software Product Line Online Tools Website. http://splot-research.org/. Accessed: 26 November 2013 (2013)
5. Fleurey, F., Haugen, Ø., Møller-Pedersen, B., Olsen, G.K., Svendsen, A., Zhang, X.: A Generic Language and Tool for Variability Modeling. Technical Report SINTEF A13505. SINTEF, Oslo, Norway (2009)
6. CVL 2 TOOL Website. http://modelbased.net/tools/cvl-2-tool/. Accessed: 3 December 2013 (2013)
7. Czarnecki K., Kim, P.: Cardinality-based Feature Modeling and Constraints: A Progress Report. In Proceeding of International Workshop on Software Factories at OOPSLA '05 (2005)
8. Haugen, Ø.: Common Variability Language (CVL). OMG Revised Submission. Available: http://omgwiki.org/variability/lib/exe/fetch.php?id=start&cache=cache&media=cvl-revised-submission.pdf (2012)
9. OMG, Object Management Group: Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification. Needham (2011)
10. Gronback, R. C.: Eclipse Modeling Project - A Domain-Specific Language Toolkit. New York: Addison-Wesley (2009)
11. Wahyudianto, Budiardjo, E. K., Zamzami, E. M.: Feature Modeling and Variability Modeling Syntactic Notation Comparison and Mapping. JCC Vol.2 No.2, January 2014. Scientific Research (2014)

