UML Based Collaborative Tool Support for Software Product Line

Suntae Kim, JeongAh Kim, and Rhan Jung

Dept. of Computer Engineering, Kangwon National University,
Sam-Cheok, South Korea
Dept. of Computer Education, Kwandong University, South Korea
fstkirril@kangwon.ac.kr
fclaral@kd.ac.kr
fjungranl@kangwon.ac.kr

1 Introduction

Software Product Line(SPL) is a paradigm to maximize software asset reusability by identifying and managing commonality and variability. Due to the massive work for building software product line architecture and product assets, many stakeholders including domain analyzers, software architects and developers should be involved. In order to facilitate the building process, standard notations and rigorous development process are necessary. Furthermore, many tools that support each activity should be integrated into one software package for seamless software development, and collaboration of participants also should be considered [4].

In this paper, we present key requirements of UML-based collaborative tool support, named UBCT, to build SPL systems and show architectural approach to building the tool. At the heart of the tool support are UML and feature models for increasing communication effectiveness among participants. They are used as an anchor to connect requirements and product line codes through maintaining traceability among features, domain and design models and codes. All artifacts are designed to be able to share via the central resource management system.

2 Traceability for SPL

To instantiate products from product lines, traceability among software architects should be maintained. Fig. 1 shows software development process composing of five activities, which is a process that our tool follows. Similar to other product line approaches[3], it begins with feature modeling for analyzing common and variable features,and then analyzes domain and design elements to realize the features with UML. In the domain and design model, the commonality and variability are represented in UML stereotypes. All design model classes are realized into classes called *feature code* which is the code segment that implements one or more features. It is surrounded by the pair of the *FEATURECODE* and *END_FEATURE_CODE* comment. Also the comment includes implementing feature list. Then, the feature code block is commented out according to the feature selection in instantiating products from product line code.

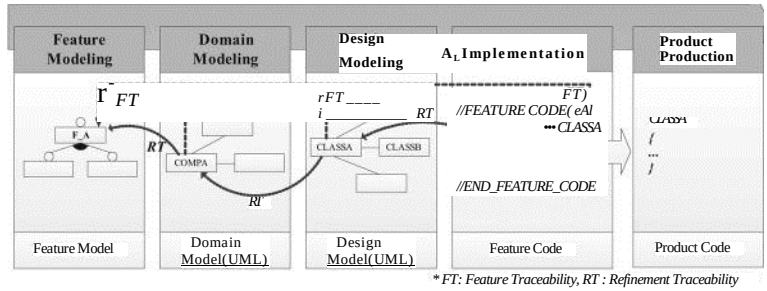

Fig. 1. Activities and Artifacts of MAP Method

The process stresses traceability among artifacts. Two types of traceability are presented in the process. One is *refinement traceability(RT)* that refers to refinement relationships between previous step elements and next step elements, indicating from which elements at the previous step the next step elements are refined. *Feature traceability(FT)* is traceability between features and elements of each artifact, enabling one to analyze the feature impact at the domain model and design model, and instantiate products from feature code according to the feature selection. Two types of traceability are properties of all UML elements, and the FT properties of the previous step element are copied into the next step elements only by defining the RT properties.

3 UML-Based Collaborative Tool Support(UBCT)

UBCT is a tool suite that supports aforementioned SPL process. To facilitate the process and support collaboration among participants, we have identified a set of key requirements as below.

- *Feature Modeling :* The tool shall allow one to model features and select features for product configuration. It also should be able to check if the selected features are not violated against feature constraints.
- *Domain and Design Modeling with UML :* The tool shall support UML modeling. The modeling tool shall show UML models in a single logical view but store them in separated physical files. It is because once one locks a single model file, others should wait for releasing the file. Thus, it is required functionality to enable parallel modeling.
- *Traceability:* Establishing traceability among the model elements and features is the crucial part for SPL system. It can be used for analyzing change impacts and generating products from product line code in accordance with feature selection.
- *Collaboration :* Source code, features and UML models shall be mutually shared and editable among participants. Based on the shared resources, traceability information shall be accordingly updated.
- *Feature Code Definition :* All features and UML models are eventually realized into feature code. Thus, the tool shall support feature code definition and view.
- *Product Production :* The tool shall produce products from the product line code by commenting out or removing the unselected feature code block.

We have been developing UBCT for realizing the requirements. It is built upon *Eclipse CDT* [1] for editing and debugging product line and product code written in

C++. We adopted two well-known Eclipse plug-ins *FMP[2]* and *Subclipse[6]*. While FMP(Feature Modeling Tool) is a tool for feature modeling and managing configurations of product features, Subclipse is intended to share file resources with collaborators by storing them in Subversion[7] source code management(SCM) system.

We adapted commercial UML modeling tool *N3Nabee* [5] for domain and design models due to stability and customizability. It allows one to define software development process and artifacts of each activity. Indeed, it supports to separate logical views and physical resources that constitute the UML model. Separating logical view and physical resources enables one to partition a huge logical model into several sections (physical resources) so that one can share each model section separately. It permits collaborators to carry out parallel UML modeling.

We are developing a component that plays a key role as a glue component that connects N3Nabee, FMP and Subclipse. It keeps information of feature code by observing feature code comments in Eclipse CDT. It obtains product configurations from FMP and then generates product codes from the product line source code according to the configuration. In addition, it provides code templates to the UML modeling tool so that the tool generates a code template from the design model.

To provide two types of traceability in collaborative environment, all related information should be separately stored in the physical resources and shared them with collaborators through central SCM. For building traceability in the tool, several UML models should be accessed and updated simultaneously. This situation always can occur data inconsistency, because one can update traceability, while others can update the UML models involved in the traceability. This problem motivates to maintain local traceability database(called *TR DB)* that keeps RT, FT of each models, physical resource location and the last modified time of the resource. The information is always updated whenever elements are modified and the tool is initially loaded. Based on the information, the tool checks inconsistency of the model elements separated in the several files. If it detects inconsistency, it alarms to users with warning message.

Acknowledgements. This research was supported by National IT Industry Promotion Agency (NIPA) under the program of Software Engineering Technologies Development and Experts Education.

References

1. Eclipse C/C++ Development Tooling. http://www.eclipse.org/cdt/.
2. University of Waterloo, Feature Modeling Plug-in. http://gsd.uwaterloo.ca/projects/fmp-plugin/.
3. **H.** Gomaa. *Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures.* Addison Wesley, 2004.
4. I. Groher, C. Schwanninger, and M. Voelter. An Integrated Aspect-Oriented Model-Driven Software Product Line Tool Suite. In *Proceedings of Intematioal Conference on Software Engineering,* pages 939-940, Leipzig, Germany, 2008.
5. N3 Soft Web Site. http://www.n3soft.co.kr/.
6. Subclipse Web Site. http://subclipse.tigris.org/.
7. Subversion Web Site. http://subversion.tigris.org/.