

HANDLING VARIABILITY IN MASS CUSTOMIZATION OF SOFTWARE FAMILY PRODUCT

Jianhong Ma¹, Runhua Tan²

¹*School of Computer Science and Software Engineering, Hebei University of Technology, TianJin, 300130, China, jh_ma@eyou.com;* ²*School of Mechanical Engineering, Hebei University of Technology, TianJin, 300130, China*

Abstract: Similar to mass customization in traditional industry, managing variability of software family products is also the key technology in software product line. In this paper, variability model and variability lifecycle are discussed. Typical approaches of modeling variability are investigated and compared in terms of the process of identifying, managing variability, model expression. The development trend of mass customization in software produce is talked about.

Key words: variability model; variability lifecycle; software product line; mass customization

1. INTRODUCTION

Similar to mass customization in traditional industry, it is the goal to develop and evolve software systems with minimal development effort and time-to-market and maximum quality in software industry. Modeling and managing variability is the key technology in software product line. Method of handling and managing variability has a direct and important effect on software reusability. Many approaches have been presented in literatures. Each of them paid more attention on some aspects but ignored others.

This project is supported by the Project of Natural Science Foundation of China under Grant No.50375045, the Project of Natural Science Foundation of Tianjin under Grant No.043802211 and the Project of Educational Foundation of HeiBei Province under Grant No.2004412

Please use the following format when citing this chapter:

Ma, Jianhong, Tan, Runhua, 2006, in International Federation for Information Processing (IFIP), Volume 207, Knowledge Enterprise: Intelligent Strategies In Product Design, Manufacturing, and Management, eds. K. Wang, Kovacs G., Wozny M., Fang M., (Boston: Springer), pp. 996-1001.

First, the variability lifecycle and variability model in each lifecycle phase are discussed. According to the variability lifecycle, typical approaches are compared in identifying, representing and implementing the variability. The shortage and development trend of handling variability in software are talked about at last.

2. VARIABILITY IN SOFTWARE PRODUCT LINE

Software variability is the ability of a software system or artifact to be changed, customized or configured for use in a particular context.

2.1 The lifecycle of variability in product line

Software product line includes two processes: application engineering and domain engineering. Variability exists throughout the whole software produce process in the product line.

1) in domain engineering phase, variability is identified, designed and implemented for reuse. It includes: variability identification, variability analysis and design and variability implementation.

2) in application engineering phase, the variability in product line is tailored and configured during developing a new member of family products. It includes: variability customization, variability configuration and variability binding.

2.2 Variability model

Variability in a software product line can be modeled at the software requirements level and the software design level. The different variability models with different roles are created at each lifecycle phase.

Variability requirement model: identifies and expresses variability abstracted from the existent family products or predicted from the future products.

Variability design model: shows how to deal with this variability in coding stage for system developer and coding engineers to implement the variability. The model is created based on variability requirement model.

Variability reuse model: directs the reuser to custom, configure or bind the variability when reuse the core asserts. There is not explicitly distinct between variability design model and reuse model, commonly, they are emerged together as design model.

3. COMPARING THE APPROACHES OF MODELING THE VARIABILITY

The approaches to model and manage the variability are classified into three main branches: feature-oriented method, objected-oriented method and integration method.

3.1 Modeling variability with object-oriented method

1992, Hassan Gomaa presented the object-oriented domain analysis and modeling method¹. It includes two major steps: first is to complete the kernel domain model, then to analyze the variations of the domain. There is no distinction between each phase of variability lifecycle. The variability is expressed in object models based on kernel model. The objects may be kernel, optional or variant objects. The objects relationships may be described as: aggregation hierarchy, supporting the IS-PART-OF relationship; generalization/specialization hierarchy, supporting the IS-A relationship; and feature/object dependencies: showing the objects required to support each feature of the domain.

The advantage of this method is that the notation of models is same as UML notation. It is easy to describe the models with UML tools. The limitation is that the reusable domain object is too difficult to be identified in practice and results in the low reusability. Inheritance mechanism makes the software product more flexibility in some ways, but it reduces the dependency between the objects or modules. There is no explicit variability reuse model, it is implicated in object models. They are expressed with natural language.

3.2 Modeling variability with feature-oriented approach

Feature means the attribute and character belonging to the system. It is the bridge that connects users and application developers. It characterizes one product from other family products. The classical feature-based method is Feature Oriented Domain Analyses(FODA) proposed in 1990 by Kang².

3.2.1 Variability identified in FODA

In FODA, the variability requirement model includes feature model, function model and ER entity. Variability is identified and described mainly in feature model. The features in feature model are classified into following categories: mandatory feature, optional feature and alternative feature.

3.2.2 Variability analyses and design in FODA

Based on variability requirement model, variability is analyzed, the architectural model(also known as a design reference model) is created. That is variability design model. It provides detailed design and component construction. It focuses on identifying concurrent processes and domain-oriented common modules, allocating the features, functions, and data objects defined in the domain models to the processes and modules. Some features have impact on the final architecture such as when to "bind" or "fix" the value of a feature, features are grouped into three classes according to the binding features such as: compile-time features, load-time features and runtime features.

In 1998, the feature-oriented reuse method(FORM) takes many non-function features into account, it concentrates on analyzing and modeling the variability in terms of capability, operating environment, domain technology, and implementation technique features.

3.2.3 Variability reuse model in FODA

The constraint relationships between variable features are used to validate whether the constraints are satisfied after tailoring and customizing the feature models for special application. The relationships between features are represented as: composition or generalization relationships, dependency relationship, exclusive and non-exclusive alternatives and mutual exclusion relationship. These relationships are modeled with UML notations.

3.3 Integrated method

Aiming at improving both feature-oriented method and object-oriented method, many approaches have been presented to integrate them together.

3.3.1 FeaturRBSE method

In 2000, FeatureRBSE⁴ integrated the feature modeling of Feature-Oriented Domain Analysis (FODA) into the approach of the Reuse-Driven Software Engineering Business (RSEB). The RSEB is a use-case driven systematic reuse process: architecture and reusable subsystems are first described by use cases and then transformed into object models that are traceable to these use cases. Variability in the RSEB is captured by structuring use case and object models using explicit variation points and

variants. Also, based on the notion of domain-oriented, it emphasizes a group of closely related applications in a domain rather than a single application, first it creates domain use case model.

Comparing with FODA, besides the compose-of relationship, mandatory or optional attributes are considered in featurRBSE, there are some extended feature relationships:

1) the alternative relationship: variation and variant features. A feature can act as a variation point (called variation point feature or vp-feature) in the model, while other features play the role of its possible variants (called variant features).

2) the binding time: attribute of vp-features. Vp-features can be bound at reuse time, i.e. when the reuser accesses the domain infrastructure to configure reusable assets for his development.

3.3.2 Extended FORM method

In 2000, kang extended FORM³ with integrating object-oriented into FORM to make the gap closely between variability identification and variability analysis and design. It presents a systematic way to identify reusable objects from feature model by linking feature categories to object categories. Variant points are described in object model that consists of objects and their relationships: aggregation, generalization and association. The relationships of the features in feature model are traced to variant points described in object model. The relationships between features (e.g., composed-of, generalization, and implemented-by) are mapped into relationships (e.g., aggregation and generalization) between objects.

3.3.3 FODM method

The feature-oriented domain modeling method (FODM)⁵ was presented by Hong Mei at 2003. This method explicitly absorbs and evolves the advantages of FODA method and FeatureRBSE method. It presents more feasibly method for reusing the requirements with use case model and feature model, the feature model is first proposed from five aspects, namely, basic structure, variation, representation mechanism, variation binding time, variation constraint mechanism and quality feature analysis. Variability constraints are described in logic, it is a guideline for reuser to tailor or bind the variability from the requirement model.

The disadvantage of this method is that there need a special tool to support these models.

4. CONCLUSION

The emphasis work of managing variability must be shifted from how to develop the core asserts to how to tailor the core asserts and how to validate its validity. There are some important questions in software product line:

- 1) How to identify the variability and design variability with the uniform notation in each model.
- 2) One variant often is reflected in different view, how to trace the variant in different model.
- 3) How to evaluate and validate the reusability, cohesion, independence of component or object with variant points.

5. REFERENCES

1. Hassan Gomaa, *An Object-Oriented domain analysis and modeling method for software reuse*. 0073-1129-1/92 1992 IEEE.
2. Kang KC, Cohen SG, Hess JA, Novak WE, Peterson AS. *Feature-Oriented domain analysis feasibility study. Technique Report*, SEI-90-TR-21, Pittsburgh: Software Engineering Institute, 1990.
3. Kwanwoo Lee¹, Kyo C. Kang¹, Wonsuk Chae¹ and ByoungWook Choi². Feature-based approach to object-oriented engineering of applications for reuse. *Softw. Pract. Exper.* 2000; 30:1025 1046.
4. Griss ML, Favaro J, d' Alessandro M. Integrating feature modeling with the RSEB. Proc. of the 5th Int'l Conf. on Software Reuse. *IEEE Computer Society*, 1998. 76-85.
5. Hong Mei, Wei Zhang, Fang Gu. A Feature Oriented Approach to Modeling and Reusing Requirements of Software Product Lines. *Proceedings of the 27th Annual International Computer Software and Applications Conference* 2003 IEEE.