HAL open science

Constructing Method Families Based on the Variability Analysis

Elena Kornyshova, Jolita Ralyte, Rebecca Deneckere

To cite this version:

Elena Kornyshova, Jolita Ralyte, Rebecca Deneckere. Constructing Method Families Based on the Variability Analysis. RCIS 2013 Forum, May 2013, Paris, France. pp.1. hal-00812071

HAL Id: hal-00812071

https://hal.science/hal-00812071v1

Submitted on 21 Dec 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Constructing Method Families Based on the Variability Analysis

Elena Kornyshova
Centre de Recherche en Informatique
Université Paris I – Panthéon Sorbonne
90, rue de Tolbiac, 75013, Paris, France
elena.kornyshova@univ-paris1.fr

Jolita Ralyté
Institute of Services Science
University of Geneva, CUI
7 route de Drize, CH-1227 Carouge, Switzerland
jolita.ralyte@unige.ch

Rébecca Deneckère
Centre de Recherche en Informatique
Université Paris I – Panthéon Sorbonne
90, rue de Tolbiac, 75013, Paris, France
rebecca.deneckere@univ-paris1.fr

Abstract— **Situational Method Engineering (SME) is constantly looking for new ways to facilitate situation-specific method construction in practice. We have developed the notion of method family to attain this goal. Our inspiration comes from Software Product Line Engineering (SPLE), which is developing solutions that can be easily adapted to a specific context and by reusing existing knowledge. In this paper, we introduce the process guiding the construction of method families based on the analysis of the methods variability.**

Keywords - Method family, variability, method family construction

I. INTRODUCTION

It is commonly agreed now that there is no standard method suitable for all software and information systems engineering situations mainly because they vary considerably from one company to another and even from one project to another. In fact, each development project has its own characteristics and needs an appropriate analysis, design and other engineering and decision supporting techniques to be arranged in an adequate way. The discipline of Situational Method Engineering (SME) claims that each development project should start with the task of method engineering where a situation-fitting method is constructed on the fly [1][2][10][11]. However, this task appears to be hard to implement in the industrial context. Despite the publication of various SME approaches (e.g. assembly-based [1][11], configuration-based [2][6], process tailoring [12], service-oriented [4][5]), SME is still not fully adopted in practice. Even though enterprises acknowledge that methods bring a real added value to their development projects, they also argue that SME processes are too complex, time consuming and too expensive.

To deal with this challenge, we take inspiration from Software Product Line Engineering (SPLE) and in particular Software Product Families [3]. SPLE is a paradigm to develop software applications using platforms and mass customization, which means that commonalities and differences in the applications of the product line have to be explicit and modeled in a common way [9]. Similarly, we claim that in SME the notion of *method family* would facilitate access to the different existing methods and would enable their combination, while at the same time simplifying the method engineering task. Therefore, our idea consists in providing software and information systems development companies with method families dedicated to their application field or purpose that could be easily configured into project-specific method lines according to the project. We propose to develop a method family engineering approach based on SME and SPLE principles.

We have introduced the concept of method family in [8]. In this paper, we focus our attention on the method family construction process based on the method variability analysis.

II. VARIABILITY IN METHOD FAMILIES

The method family concept aims at responding to the question of how to organize method components into a multi-process and configurable method. Most of the SME approaches are based on the assumption that a method has to be composed at time of a specific project "on the fly" [1][2][10][11]. The concept of method family addresses this issue in a different way – method components from the same application field and having the same or similar main usage are organized into a method family before any concrete project. That allows to reduce the complexity in managing the vast variety of methods and to limit the project-specific method construction to the method configuration task during the method application phase. To develop this solution, we use the idea of variants and variation points from SPLE that we apply to the domain of SME.

A *method family* [8] represents an organization of several method components, from different methods, into a multi-method dedicated to a particular engineering field or purpose. The method family adaptation, or rather configuration, to a particular project is made by selecting an appropriate method line inside this family. A method line includes all common method components of the method family and some variable method components selected for the given project based on the project characteristics. Regarding to the variability expression, a method component in a method family is considered similarly to a variant in a software product line. In SPLE, variants are artifacts available in a given variation point [9]. Applied to the method family notion, variants are method components available in a given situation (or in a given variation point). Therefore, during the method family configuration, each obtained method line is composed of a set of variant components. The differentiation between common and variable method components allows to evaluate the degree of variability of a method family and helps to configure method lines.

In order to define the way of representing variability in method families, we have used the ideas of the Orthogonal Variability Model (OVM) proposed by Pohl et al. [9]. OVM is "a model that defines the variability of a software product line.

It relates the variability defined to other software development models such as feature models, use case models, design models, component models, and test models" [9]. We use the following elements of this model to deal with variability in method families: variation point, variability dependency, and constraint variability. The situation containing several variants (in our case method components) is called a *variation point*. The relationship between the variation point and variants defines the *variability dependency*. It induces a dependency restriction, as a variant may be optional or mandatory within a given variation point (See [8] for the detailed description of the method family meta-model).

III. Method Family Construction Process

Following the SME recommendations and principles, our approach for method family construction is based on the reuse of method components extracted from existing methods and on their variability analysis. In particular, this approach aims to support the method family construction from a collection of methods belonging to the same application domain and having similar objectives. The construction process is depicted in the following figure.

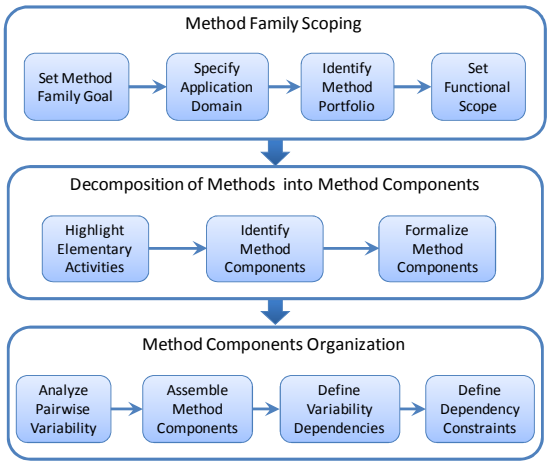

Figure 1. Process of the method family construction

Method Family Scoping. Different methods may be federated into a family only if they satisfy the property of "teleological unity". This property characterizes the similarity of different methods according to their usage expressed through their intentions [7]. Our method family scoping phase incorporates ideas from [1], [7] and [2] and includes four steps: (1) defining the method family goal (e.g. a support for decision making), (2) specifying its application domain (e.g. information systems engineering), (3) identifying method portfolio (the spectrum of methods fitting the method family goal and the application domain), and (4) setting its functional scope (e.g. from requirements to design).

Decomposition of Methods into Method Components. The method components are extracted from the existing methods, composing the method portfolio defined in the previous phase. The decomposition of methods is process-driven and includes three steps: (1) highlighting elementary activities in the method, (2) identifying method components, and (3) formalizing them following a selected method engineering approach.

Components Organization into a Family. The last phase consists in organizing all previously defined method components within a method family. It is based on (i) the assembly-based SME approach [11] and (ii) the variability analysis with similarity measures. This phase is composed of four steps: (1) pairwise variability analysis (defining the variability of method components based on similarity measures), (2) assembly of method components by using assembly techniques [11], (3) definition of variability dependencies (optional or mandatory) and (4) definition of dependency constraints between method components (exclusive or requires each other).

IV. Conclusion

Method families bring together a set of similar but different method components. Their main aim is to facilitate method reuse and adaptation in practice. Instead of creating a project-specific method from scratch as suggested by many SME approaches, the method family user will configure a method line by selecting method components from the family. His/her method engineering effort will be significantly reduced while the offer of method component will be increased.

In this paper we have introduced a process model for the construction of method families. The approach is tightly related to the variability notion underlying the method family concept. The combination of the variability analysis issues from the SPLE field with the assembly-based SME techniques forms the basis for our method family construction process. This method family construction approach will be extended and tested in various case studies and projects.

References

[1] S. Brinkkemper, M. Saeki, and F. Harmsen, "Assembly Techniques for Method Engineering", CAiSE 1998, LNCS 1413, pp. 381-400, 1998.

[2] T. Bucher, M. Klesse, S. Kurpjuweit, and R. Winter, "Situational Method Engineering. On the Differentiation of Context and Project Type", ME 07, Vol. 244, Springer, Boston, pp. 33-48, 2007.

[3] P. Clements and L. Northrop, Software Product Lines: Practices and Patterns, Addison-Wesley Longman Publishing Co., USA, 2001.

[4] R. Deneckère, A. Iacovelli, E. Kornyshova, and C. Souveyet, "From Method Fragments to Method Services", EMMSAD, pp. 80-96, ceur-ws.org, Vol.337, 2008.

[5] G. Guzélian and C. Cauvet, "SO2M: Towards a Service-Oriented Approach for Method Engineering", IKE'07, Las Vegas, USA, 2007.

[6] F. Karlsson and P.J. Ågerfalk, "Method Configuration: Adapting to Situational Characteristics While Creating Reusable Assets", Information and Software Technology, Vol. 46(9): 619–633, 2004.

[7] E. Kornyshova, "MADISE: Method Engineering-based Approach for Enhancing Decision-Making in Information Systems Engineering", PhD thesis, University of Paris 1 Panthéon-Sorbonne, Paris, France, 2011.

[8] E. Kornyshova, R. Deneckère, and C. Rolland, "Method Families Concept: Application to Decision-Making Methods", EMMSAD, pp. 413–427, LNBIP 81, 2011.

[9] K. Pohl, G. Böckle, F. Van der Linden, Software Product Line Engineering: Foundations, Principles, and Techniques, Springer Verlag, 2005.

[10] J. Ralyté, R. Deneckère and C. Rolland, "Towards a Generic Model for Situational Method Engineering", CAiSE 2003, pp. 95-110, LNCS 2681, 2003.

[11] J. Ralyté and C. Rolland, "An Assembly Process Model for Method Engineering", CAISE 2001, pp. 267-283, LNCS 2068, Springer, 2001.

[12] M. Rossi, B. Ramesh, K. Lyytinen, and J-P. Tolvanen, "Managing evolutionary method engineering by method rationale", Journal of the Association for Information Systems, Vol. 5(9), pp. 356-391, 2004.

View publication stats