

Reengineering Software Product Variants into Software Product Line: REVPLINE Approach

Ra'Fat Ahmad Al-Msie'Deen, Abdelhak-Djamel Seriai, Marianne Huchard

▶ **To cite this version:**

Ra'Fat Ahmad Al-Msie'Deen, Abdelhak-Djamel Seriai, Marianne Huchard. Reengineering Software Product Variants into Software Product Line: REVPLINE Approach. Lambert Academic Publishing (LAP), 120 p., 2014, 978-3-659-51125-7. lirmm-00981473

HAL Id: lirmm-00981473

https://hal-lirmm.ccsd.cnrs.fr/lirmm-00981473

Submitted on 22 Apr 2014

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.

In order to migrate software product variants that are considered similar into a Software Product Line (SPL), it is essential to identify the mandatory and optional features between the product variants. To exploit existing software variants and build a SPL, a feature model of this SPL must be built as a first step. To do so, it is necessary to mine optional and mandatory features from the source code of the software variants. Thus, we propose in this book, a new approach to mine features and feature models from the object-oriented source code of a set of software variants, based on Formal Concept Analysis and Latent Semantic Indexing. To validate our approach, we applied it to ArgoUML and Mobile media case studies. The results of this evaluation validate the relevance and the performance of our proposal, as most of the features were correctly identified.



Ra'Fat AL-Msie'Deen

Ra'Fat Al-Msie'Deen received Bachelor of computer science in September 17, 2007 from Al - Hussein Bin Talal University in Jordan, and he received a Master of Science (Information Technology) in March 28, 2009 from University Utara Malaysia. His research interest includes Software Product Line Engineering and Formal Concept Analysis.

978-3-659-51125-7

Software Product Line In Action-REVPLINE

AL-Msie'Deen, Seriai, Huchard



Ra'Fat AL-Msie'Deen
Abdelhak-Djamel Seriai
Marianne Huchard

Reengineering Software Product Variants Into Software Product Line

REVPLINE Approach

LAP LAMBERT Academic Publishing

Ra'Fat AL-Msie'Deen
Abdelhak-Djamel Seriai
Marianne Huchard

Reengineering Software Product Variants Into Software Product Line

Ra'Fat AL-Msie'Deen
Abdelhak-Djamel Seriai
Marianne Huchard

Reengineering Software Product Variants Into Software Product Line

REVPLINE Approach

LAP LAMBERT Academic Publishing

Impressum / Imprint
Bibliografische Information der Deutschen Nationalbibliothek: Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.
Alle in diesem Buch genannten Marken und Produktnamen unterliegen warenzeichen-, marken- oder patentrechtlichem Schutz bzw. sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Inhaber. Die Wiedergabe von Marken, Produktnamen, Gebrauchsnamen, Handelsnamen, Warenbezeichnungen u.s.w. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutzgesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Bibliographic information published by the Deutsche Nationalbibliothek: The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.
Any brand names and product names mentioned in this book are subject to trademark, brand or patent protection and are trademarks or registered trademarks of their respective holders. The use of brand names, product names, common names, trade names, product descriptions etc. even without a particular marking in this works is in no way to be construed to mean that such names may be regarded as unrestricted in respect of trademark and brand protection legislation and could thus be used by anyone.

Coverbild / Cover image: www.ingimage.com

Verlag / Publisher:
LAP LAMBERT Academic Publishing
ist ein Imprint der / is a trademark of
OmniScriptum GmbH & Co. KG
Heinrich-Böcking-Str. 6-8, 66121 Saarbrücken, Deutschland / Germany
Email: info@lap-publishing.com

Herstellung: siehe letzte Seite /
Printed at: see last page
ISBN: 978-3-659-51125-7

Copyright © 2013 OmniScriptum GmbH & Co. KG
Alle Rechte vorbehalten. / All rights reserved. Saarbrücken 2013

Reengineering Software Product Variants into Software Product Line: REVPLINE Approach

R. AL-Msie'Deen, A.-D. Seriai, and M. Huchard

Contents

1 REVPLINE Approach **9**

1.1 Abstract 9

1.2 Introduction 10

1.3 Background 13

1.3.1 Software Product Line Engineering (SPLE) 13

1.3.2 Software Product Variants (SPVs) 25

1.3.3 Formal Concept Analysis (FCA) 28

1.3.4 Latent Semantic Indexing (LSI) 40

1.3.5 Structural similarity 55

1.4 Approach Basics 56

1.4.1 Goal and Core Assumptions 56

1.4.2 Features versus Object-oriented Building Elements: the Mapping Model 58

1.4.3 Object-oriented source code variation 62

1.4.4 An Illustrative Example 63

1.5 The Feature Mining Process 64

1.5.1 Identifying the Common Block and Blocks of Variation . 65

1.5.2 Identifying Atomic Blocks 68

1.6 Experimentation 72

1.6.1 Case studies 73

1.6.2 Implementation 74

1.6.3 Mining common block and blocks of variation using FCA 75
1.6.4 Mining common atomic blocks and atomic blocks of variation using FCA and LSI 76
1.6.5 Reverse engineering feature models from program configurations 78
1.7 Related Work 81
1.7.1 Feature location approaches 81
1.7.2 Reverse engineering FM 87
1.8 Conclusion and Perspectives 92
1.9 Acknowledgements 94

List of Figures

1.1 Multiple versions of software. 10
1.2 From software variant artifacts to a feature model (inspired by [1]). 12
1.3 Mining features from software product variants source code (inspired by [2]). 12
1.4 Overview of an engineering process for software product lines [3]. 15
1.5 Costs for developing n kinds of systems as single systems compared to SPLE [4]. 17
1.6 Cell phone SPL feature model [5]. 22
1.7 Video on demand SPL feature model. 25
1.8 The clone-and-own approach. 27
1.9 The concept lattice for the context in Table 1.4. 30
1.10 The concept lattice for the context in Table 1.4. 31
1.11 The concept lattice for the formal context of Table 1.4. 32
1.12 The AOC-poset for the formal context of Table 1.4. 33
1.13 The AOC-poset for the formal context of Table 1.5 [6]. 34
1.14 The AOC-poset for the formal context of Table 1.6. 36
1.15 The concept lattice for the formal context of Table 1.6. 38
1.16 The concept lattice for the formal context of Table 1.7. 40
1.17 Graphical representation of the singular value decomposition for TDM [7]). 48

1.18 Reduced singular value decomposition of the term-document matrix [7]). 48
1.19 Commonalities and variability cross software product variants source code. 57
1.20 The common block and blocks of variation. 59
1.21 The common atomic blocks and atomic blocks of variation. . . . 59
1.22 OBE to feature mapping model. 60
1.23 An illustrative example. 61
1.24 Object-oriented source code variation. 63
1.25 Text editor feature model. 64
1.26 The feature mining process. 65
1.27 The AOC-poset for the formal context of Table 1.23. 66
1.28 The AOC-poset for the formal context of Table 1.24. 67
1.29 Atomic blocks mined from *Concept_2*. 71
1.30 The mined features from text editor variants. 72
1.31 OBEs of software as XML file. 74
1.32 Visualization of textual similarity between OBEs of (*Concept_2*) in Figure 1.27. 75
1.33 Part of *activity diagram* feature implementation from ArgoUML SPL. 78
1.34 Mining feature model from software product variants source code (inspired by [8]). 79
1.35 The mined feature model for ArgoUML-SPL. 80
1.36 The mined feature model for Mobile media software variants. . . 80
1.37 *OBE to Feature* mapping model. 82
1.38 Comparing the feature location approaches in a collection of software product variants via FM. 85
1.39 Comparing the feature location approaches in a collection of software product variants via concept lattice. 86

1.40 Reverse engineering FM techniques and their input artifacts as FM. 90
1.41 Reverse engineering FM techniques and their input artifacts as concept lattice. 91
1.42 Extraction of a product by selecting its features from FM. . . . 93

List of Tables

1.1 Feature sets of cell phone SPL [5]. 24
1.2 Feature sets of video on demand SPL [9] [1]. 25
1.3 Animals and their characteristics. 28
1.4 A formal context for animals and their characteristics ($O \times A$). 29
1.5 A formal context describing drawing shapes software variants [6]. 35
1.6 A formal context describing bank systems by construction primitives [10]. 37
1.7 A formal context describing wiki systems by characteristics [11]. 39
1.8 Two examples of synonymy. 43
1.9 An example of polysemy. 43
1.10 LSI example: technical memo example [7]. 46
1.11 Term by document matrix. 47
1.12 The 12-term by 9-document matrix [7]. 49
1.13 $SVD-$ The T (term) matrix. 50
1.14 $SVD-$ The singular values (S) matrix. 50
1.15 $SVD-$ The D (document) matrix. 50
1.16 The $TDM = TSD'(T \cdot S \cdot D)$. 50
1.17 The TDM_k matrix. 51
1.18 The term-document matrix and term-query matrix. 52
1.19 The T, S and D matrices. 52
1.20 The T_k, S_k and D_k matrices (TSD'). 53

1.21 The new query vector coordinates in the reduced 2-dimensional space. 54
1.22 Object-oriented building elements for software. 62
1.23 The formal context for the text editor variants. 65
1.24 The formal context for the word editor variants. 67
1.25 The similarity matrix of (*Concept_2*) in Figure 1.27. 70
1.26 Formal context of *Concept_2*. 71
1.27 Mobile media software product variants. 73
1.28 ArgoUML software product variants. 73
1.29 Feature location in ArgoUML software variants. 75
1.30 Junction location in ArgoUML software variants. 76
1.31 Features mined from Mobile media and ArgoUML softwares. . . 77
1.32 ArgoUML product configurations. 79
1.33 Comparing the feature location approaches in a collection of software product variants. 84
1.34 Reverse engineering FM techniques and its input artifacts. . . . 89

1

REVPLINE Approach

1.1 Abstract

In order to migrate software product variants that are considered similar into a Software Product Line (SPL), it is essential to identify the mandatory and optional features between the product variants. To exploit existing software variants and build a SPL, a feature model of this SPL must be built as a first step. To do so, it is necessary to mine optional and mandatory features from the source code of the software variants. Thus, we propose in this book, a new approach to mine features and feature models from the object-oriented source code of a set of software variants, based on Formal Concept Analysis and Latent Semantic Indexing. To validate our approach, we applied it to ArgoUML and Mobile media case studies. The results of this evaluation validate the relevance and the performance of our proposal as most of the features were correctly identified.

Keywords: REVPLINE approach, Reengineering, Feature mining, Feature Model, Software Product Line Engineering, Software product variants, Structural similarity, Lexical similarity, Formal Concept Analysis, Latent Semantic Indexing, Code dependencies.