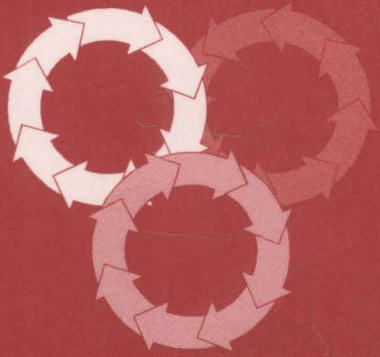
Henk Obbink
Klaus Pohl (Eds.)

LNCS 3714

Software Product Lines

**9th International Conference, SPLC 2005
Rennes, France, September 2005
Proceedings**

SPLC 2005



Springer

Henk Obbink Klaus Pohl (Eds.)

Software Product Lines

9th International Conference, SPLC 2005
Rennes, France, September 26-29, 2005
Proceedings

Volume Editors

Henk Obbink
Philips Research Laboratories Eindhoven
Prof. Holstlaan 4, 5656 JA Eindhoven, The Netherlands
E-mail: Henk.Obbink@philips.com

Klaus Pohl
Universität Duisburg-Essen, Campus Essen
Institut für Informatik and Wirtschaftsinformatik
Software Systems Engineering
Schuetzenbahn 70, 45117 Essen, Germany
E-mail: Pohl@sse.uni-essen.de

Library of Congress Control Number: 2005932544

CR Subject Classification (1998): D.2, K.4.3, K.6

ISSN 0302-9743
ISBN-10 3-540-28936-4 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-28936-4 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11554844 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 3714

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

Preface

With SPLC 2005 we celebrated the formation of a new conference series, the *International* Software Product Line Conference (SPLC) which results from the "unification" of the former series of three SPLC (Software Product Line) Conferences launched in 2000 in the USA, and the former series of five PFE (Product Family Engineering) Workshops started in 1996 in Europe.

SPLC is now *the premier forum* for the growing community of software product line practitioners, researchers, and educators. SPLC offers a unique opportunity to present and discuss the most recent experiences, ideas, innovations, trends, and concerns in the area of *software product line engineering* and to build an international network of product line champions. An international SPLC Steering Committee has been established and it is the wish of this committee that from 2005 on, the SPLC conference will be held yearly in Europe, America, or Asia. The technical program of SPLC 2005 included.

- two keynotes from David Weiss (Avaya, USA) and Jan Bosch (Nokia, Finland), both leading experts with academic and industrial insights;
- 17 full and 3 short research papers organized around the following themes: feature modeling, re-engineering, strategies, validation, scoping and architecture, and product derivation;
- eight experience reports describing commercial application of product line practices;
- two panels focused on special topics in product line practice and product line research;
- tool demonstrations;
- a Hall of Fame session that continued the SPLC tradition in a slightly revised format.

In addition, the technical program was preceded by a tutorial and workshop day that included ten half-day tutorials presented by well-recognized experts and five workshops on specific areas of product line research.

The preparation of this programme would not have been possible without the help and support of many individuals. The role of the Program Committee was central in the achievement of this high-quality programme. We are indebted to each PC member for his or her commitment in reviewing the papers, participating in electronic consensus discussions and, finally, in actively taking part in the PC meeting, which was held in Essen on May 24, 2005.

Thanks also to the Organizing Committee and in particular to Jean-Marc Jézéquel for his continuous support at all stages and for making it possible to host SPLC 2005 in the beautiful city of Rennes. Most especially, we would like to thank all those who submitted their work to SPLC 2005. Without their willingness to publish and share their work SPLC 2005 would not have been possible.

June 2005 Henk Obbink and Klaus Pohl

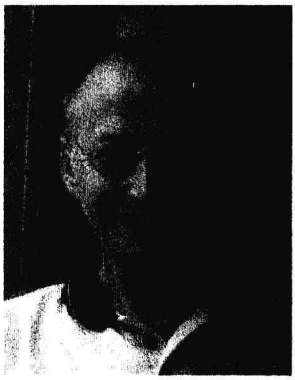
Organization

General Co-chairs

Linda Northrop

Frank van der Linden

Program Co-chairs



Henk Obbink

Klaus Pohl

Organizing Committee

General Chairs

Frank van der Linden
Philips, The Netherlands

Linda Northrop
Software Engineering Institute, USA

Program Chairs	Henk Obbink Philips, The Netherlands
	Klaus Pohl University of Duisburg-Essen, Germany
Local Organization Chair	Jean-Marc Jézéquel INRIA, France
Workshop Chairs	Svein Hallsteinsen SINTEF, Norway
	Benoit Baudry INRIA, France
Panel Chair	Charles W. Krueger BigLever Software, USA
Tool Demonstration Chair	Dirk Muthig Fraunhofer IESE, Germany
Hall of Fame	Paul Clements Software Engineering Institute, USA

Program Committee

Pierre America	Philips, The Netherlands
Joe Baumann	HP, USA
Sergio Bandinelli	ESI, Spain
Len Bass	SEI, USA
Günter Böckle	Siemens, Germany
Manfred Broy	TU Munich, Germany
Paul Clements	SEI, USA
Krzysztof Czarnecki	University of Waterloo, Canada
Juan Carlos Dueñas	University of Madrid, Spain
Birgit Geppert	Avaya, USA
Stefania Gnesi	IEI-CNR, Italy
Andre van der Hoek	University of California, USA
Kyo C. Kang	University of Pohang, Korea
Kari Känsälä	Nokia, Finland
Tomoji Kishi	JAIST, Japan
Stefan Kowalewski	RWTH Aachen, Germany
Philippe Kruchten	University of British Columbia, Canada
Charles W. Krueger	BigLever Software, USA

Tomi Männistö	HUT, Finland
John McGregor	Clemson University, USA
Nenad Medvidović	USC, USA
Dirk Muthig	Fraunhofer IESE, Germany
Robert Nord	Siemens, USA
Henk Obbink	Philips, The Netherlands (Chair)
Rob van Ommering	Philips, The Netherlands
Klaus Pohl	University of Duisburg-Essen, Germany (Chair)
Serge Salicki	Thales, France
Thomas Stauner	BMW, Germany
Steffen Thiel	Bosch, Germany
Martin Verlage	Market Maker, Germany
Matthias Weber	DaimlerChrysler, Germany

Additional Reviewers

Michal Antkiewicz
Jose L. Arciniegas
Somo Banerjee
Gerd Beneken
Jon Bentley
Ali Botorabi
Antonio Bucchiarone
Rodrigo Cerón
Alessandro Fantechi
Andreas Fleischmann
Gregory de Fombelle
Claudia Fritsch
Lars Geyer
Piergiorgio Di Giacomo
Pedro Gutierrez
Anna Ioschpe
Vladimir Jakobac
Eric Jouenne
Reinhard Klemm
Giuseppe Lami
Xabier Larrucea
Sam Malek
Jason Xabier Mansell
Chris Mattmann
Franco Mazzanti
Holt Mebane
Audris Mockus
David Morera
Ana R. Moya
Veronique Normand
Mark-Oliver Reiser
Laurent Rioux
Frank Roessler
Roshanak Roshandel
Kathrin Scheidemann
Tilman Seifert
Chiyoung Seo
Iratxe Gomez Susaeta
David Woollard

Lecture Notes in Computer Science

For information about Vols. 1–3597

please contact your bookseller or Springer

Vol. 3728: V. Paliouras, J. Vounckx, D. Verkest (Eds.), Integrated Circuit and System Design. XV, 753 pages. 2005.

Vol. 3718: V.G. Ganzha, E.W. Mayr, E.V. Vorozhtsov (Eds.), Computer Algebra in Scientific Computing. XII, 502 pages. 2005.

Vol. 3714: H. Obbink, K. Pohl (Eds.), Software Product Lines. XIII, 235 pages. 2005.

Vol. 3710: M. Barni, I. Cox, T. Kalker, H.J. Kim (Eds.), Digital Watermarking. XII, 485 pages. 2005.

Vol. 3703: F. Fages, S. Soliman (Eds.), Principles and Practice of Semantic Web Reasoning. VIII, 163 pages. 2005.

Vol. 3702: B. Beckert (Ed.), Automated Reasoning with Analytic Tableaux and Related Methods. XIII, 343 pages. 2005. (Subseries LNAI).

Vol. 3698: U. Furbach (Ed.), KI 2005: Advances in Artificial Intelligence. XIII, 409 pages. 2005. (Subseries LNAI).

Vol. 3697: W. Duch, J. Kacprzyk, E. Oja, S. Zadrożny (Eds.), Artificial Neural Networks: Formal Models and Their Applications - ICANN 2005, Part II. XXXII, 1045 pages. 2005.

Vol. 3696: W. Duch, J. Kacprzyk, E. Oja, S. Zadrożny (Eds.), Artificial Neural Networks: Biological Inspirations - ICANN 2005, Part I. XXXI, 703 pages. 2005.

Vol. 3691: A. Gagalowicz, W. Philips (Eds.), Computer Analysis of Images and Patterns. XIX, 865 pages. 2005. (Subseries LNAI).

Vol. 3690: M. Pechoucek, P. Petta, L.Z. Varga (Eds.), Multi-Agent Systems and Applications IV. XVII, 667 pages. 2005. (Subseries LNAI).

Vol. 3687: S. Singh, M. Singh, C. Apte, P. Perner (Eds.), Pattern Recognition and Image Analysis, Part II. XXV, 809 pages. 2005.

Vol. 3686: S. Singh, M. Singh, C. Apte, P. Perner (Eds.), Pattern Recognition and Data Mining, Part I. XXVI, 689 pages. 2005.

Vol. 3684: R. Khosla, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part IV. LXXIX, 933 pages. 2005. (Subseries LNAI).

Vol. 3683: R. Khosla, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part III. LXXX, 1397 pages. 2005. (Subseries LNAI).

Vol. 3682: R. Khosla, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part II. LXXIX, 1371 pages. 2005. (Subseries LNAI).

Vol. 3681: R. Khosla, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part I. LXXX, 1319 pages. 2005. (Subseries LNAI).

Vol. 3678: A. McLysaght, D.H. Huson (Eds.), Comparative Genomics. VIII, 167 pages. 2005. (Subseries LNBI).

Vol. 3677: J. Dittmann, S. Katzenbeisser, A. Uhl (Eds.), Communications and Multimedia Security. XIII, 360 pages. 2005.

Vol. 3675: Y. Luo (Ed.), Cooperative Design, Visualization, and Engineering. XI, 264 pages. 2005.

Vol. 3674: W. Jonker, M. Petković (Eds.), Secure Data Management. X, 241 pages. 2005.

Vol. 3672: C. Hankin, I. Siveroni (Eds.), Static Analysis. X, 369 pages. 2005.

Vol. 3671: S. Bressan, S. Ceri, E. Hunt, Z.G. Ives, Z. Bellahsène, M. Rys, R. Unland (Eds.), Database and XML Technologies. X, 239 pages. 2005.

Vol. 3670: M. Bravetti, L. Kloul, G. Zavattaro (Eds.), Formal Techniques for Computer Systems and Business Processes. XIII, 349 pages. 2005.

Vol. 3665: K. S. Candan, A. Celentano (Eds.), Advances in Multimedia Information Systems. X, 221 pages. 2005.

Vol. 3664: C. Türker, M. Agosti, H.-J. Schek (Eds.), Peer-to-Peer, Grid, and Service-Orientation in Digital Library Architectures. X, 261 pages. 2005.

Vol. 3663: W.G. Kropatsch, R. Sablatnig, A. Hanbury (Eds.), Pattern Recognition. XIV, 512 pages. 2005.

Vol. 3662: C. Baral, G. Greco, N. Leone, G. Terracina (Eds.), Logic Programming and Nonmonotonic Reasoning. XIII, 454 pages. 2005. (Subseries LNAI).

Vol. 3661: T. Panayiotopoulos, J. Gratch, R. Aylett, D. Ballin, P. Olivier, T. Rist (Eds.), Intelligent Virtual Agents. XIII, 506 pages. 2005. (Subseries LNAI).

Vol. 3660: M. Beigl, S. Intille, J. Rekimoto, H. Tokuda (Eds.), UbiComp 2005: Ubiquitous Computing. XVII, 394 pages. 2005.

Vol. 3659: J.R. Rao, B. Sunar (Eds.), Cryptographic Hardware and Embedded Systems – CHES 2005. XIV, 458 pages. 2005.

Vol. 3658: V. Matoušek, P. Mautner, T. Pavelka (Eds.), Text, Speech and Dialogue. XV, 460 pages. 2005. (Subseries LNAI).

Vol. 3655: A. Aldini, R. Gorrieri, F. Martinelli (Eds.), Foundations of Security Analysis and Design III. VII, 273 pages. 2005.

Vol. 3654: S. Jajodia, D. Wijesekera (Eds.), Data and Applications Security XIX. X, 353 pages. 2005.

Vol. 3653: M. Abadi, L. de Alfaro (Eds.), CONCUR 2005 – Concurrency Theory. XIV, 578 pages. 2005.

Vol. 3652: A. Rauber, S. Christodoulakis, A M. Tjoa (Eds.), Research and Advanced Technology for Digital Libraries. XVIII, 545 pages. 2005.

Vol. 3649: W.M.P. van der Aalst, B. Benatallah, F. Casati, F. Curbera (Eds.), Business Process Management. XII, 472 pages. 2005.

Vol. 3648: J.C. Cunha, P.D. Medeiros (Eds.), Euro-Par 2005 Parallel Processing. XXXVI, 1299 pages. 2005.

Vol. 3646: A. F. Famili, J.N. Kok, J.M. Peña, A. Siebes, A. Feelders (Eds.), Advances in Intelligent Data Analysis VI. XIV, 522 pages. 2005.

Vol. 3645: D.-S. Huang, X.-P. Zhang, G.-B. Huang (Eds.), Advances in Intelligent Computing, Part II. XIII, 1010 pages. 2005.

Vol. 3644: D.-S. Huang, X.-P. Zhang, G.-B. Huang (Eds.), Advances in Intelligent Computing, Part I. XXVII, 1101 pages. 2005.

Vol. 3642: D. Ślezak, J. Yao, J.F. Peters, W. Ziarko, X. Hu (Eds.), Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, Part II. XXIII, 738 pages. 2005. (Subseries LNAI).

Vol. 3641: D. Ślezak, G. Wang, M. Szczuka, I. Düntsch, Y. Yao (Eds.), Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, Part I. XXIV, 742 pages. 2005. (Subseries LNAI).

Vol. 3639: P. Godefroid (Ed.), Model Checking Software. XI, 289 pages. 2005.

Vol. 3638: A. Butz, B. Fisher, A. Krüger, P. Olivier (Eds.), Smart Graphics. XI, 269 pages. 2005.

Vol. 3637: J. M. Moreno, J. Madrenas, J. Cosp (Eds.), Evolvable Systems: From Biology to Hardware. XI, 227 pages. 2005.

Vol. 3636: M.J. Blesa, C. Blum, A. Roli, M. Sampels (Eds.), Hybrid Metaheuristics. XII, 155 pages. 2005.

Vol. 3634: L. Ong (Ed.), Computer Science Logic. XI, 567 pages. 2005.

Vol. 3633: C. Bauzer Medeiros, M. Egenhofer, E. Bertino (Eds.), Advances in Spatial and Temporal Databases. XIII, 433 pages. 2005.

Vol. 3632: R. Nieuwenhuis (Ed.), Automated Deduction – CADE-20. XIII, 459 pages. 2005. (Subseries LNAI).

Vol. 3631: J. Eder, H.-M. Haav, A. Kalja, J. Penjam (Eds.), Advances in Databases and Information Systems. XIII, 393 pages. 2005.

Vol. 3630: M.S. Capcarrere, A.A. Freitas, P.J. Bentley, C.G. Johnson, J. Timmis (Eds.), Advances in Artificial Life. XIX, 949 pages. 2005. (Subseries LNAI).

Vol. 3629: J.L. Fiadeiro, N. Harman, M. Roggenbach, J. Rutten (Eds.), Algebra and Coalgebra in Computer Science. XI, 457 pages. 2005.

Vol. 3628: T. Gschwind, U. Aßmann, O. Nierstrasz (Eds.), Software Composition. X, 199 pages. 2005.

Vol. 3627: C. Jacob, M.L. Pilat, P.J. Bentley, J. Timmis (Eds.), Artificial Immune Systems. XII, 500 pages. 2005.

Vol. 3626: B. Ganter, G. Stumme, R. Wille (Eds.), Formal Concept Analysis. X, 349 pages. 2005. (Subseries LNAI).

Vol. 3625: S. Kramer, B. Pfahringer (Eds.), Inductive Logic Programming. XIII, 427 pages. 2005. (Subseries LNAI).

Vol. 3624: C. Chekuri, K. Jansen, J.D.P. Rolim, L. Trevisan (Eds.), Approximation, Randomization and Combinatorial Optimization. XI, 495 pages. 2005.

Vol. 3623: M. Liśkiewicz, R. Reischuk (Eds.), Fundamentals of Computation Theory. XV, 576 pages. 2005.

Vol. 3622: V. Vene, T. Uustalu (Eds.), Advanced Functional Programming. IX, 359 pages. 2005.

Vol. 3621: V. Shoup (Ed.), Advances in Cryptology – CRYPTO 2005. XI, 568 pages. 2005.

Vol. 3620: H. Muñoz-Avila, F. Ricci (Eds.), Case-Based Reasoning Research and Development. XV, 654 pages. 2005. (Subseries LNAI).

Vol. 3619: X. Lu, W. Zhao (Eds.), Networking and Mobile Computing. XXIV, 1299 pages. 2005.

Vol. 3618: J. Jedrzejowicz, A. Szepietowski (Eds.), Mathematical Foundations of Computer Science 2005. XVI, 814 pages. 2005.

Vol. 3617: F. Roli, S. Vitulano (Eds.), Image Analysis and Processing – ICIAP 2005. XXIV, 1219 pages. 2005.

Vol. 3615: B. Ludäscher, L. Raschid (Eds.), Data Integration in the Life Sciences. XII, 344 pages. 2005. (Subseries LNBI).

Vol. 3614: L. Wang, Y. Jin (Eds.), Fuzzy Systems and Knowledge Discovery, Part II. XLI, 1314 pages. 2005. (Subseries LNAI).

Vol. 3613: L. Wang, Y. Jin (Eds.), Fuzzy Systems and Knowledge Discovery, Part I. XLI, 1334 pages. 2005. (Subseries LNAI).

Vol. 3612: L. Wang, K. Chen, Y. S. Ong (Eds.), Advances in Natural Computation, Part III. LXI, 1326 pages. 2005.

Vol. 3611: L. Wang, K. Chen, Y. S. Ong (Eds.), Advances in Natural Computation, Part II. LXI, 1292 pages. 2005.

Vol. 3610: L. Wang, K. Chen, Y. S. Ong (Eds.), Advances in Natural Computation, Part I. LXI, 1302 pages. 2005.

Vol. 3608: F. Dehne, A. López-Ortiz, J.-R. Sack (Eds.), Algorithms and Data Structures. XIV, 446 pages. 2005.

Vol. 3607: J.-D. Zucker, L. Saitta (Eds.), Abstraction, Reformulation and Approximation. XII, 376 pages. 2005. (Subseries LNAI).

Vol. 3606: V. Malyshkin (Ed.), Parallel Computing Technologies. XII, 470 pages. 2005.

Vol. 3605: Z. Wu, M. Guo, C. Chen, J. Bu (Eds.), Embedded Software and Systems. XIX, 610 pages. 2005.

Vol. 3604: R. Martin, H. Bez, M. Sabin (Eds.), Mathematics of Surfaces XI. IX, 473 pages. 2005.

Vol. 3603: J. Hurd, T. Melham (Eds.), Theorem Proving in Higher Order Logics. IX, 409 pages. 2005.

Vol. 3602: R. Eigenmann, Z. Li, S.P. Midkiff (Eds.), Languages and Compilers for High Performance Computing. IX, 486 pages. 2005.

Vol. 3599: U. Aßmann, M. Aksit, A. Rensink (Eds.), Model Driven Architecture. X, 235 pages. 2005.

Vol. 3598: H. Murakami, H. Nakashima, H. Tokuda, M. Yasumura, Ubiquitous Computing Systems. XIII, 275 pages. 2005.

Table of Contents

Keynotes

Next Generation Software Product Line Engineering
David M. Weiss 1

Software Product Families in Nokia
Jan Bosch 2

Feature Modelling

Feature Models, Grammars, and Propositional Formulas
Don Batory 7

Using Product Sets to Define Complex Product Decisions
Mark-Oliver Reiser, Matthias Weber 21

The PLUSS Approach - Domain Modeling with Features, Use Cases and Use Case Realizations
Magnus Eriksson, Jürgen Börstler, Kjell Borg 33

Re-engineering

Feature-Oriented Re-engineering of Legacy Systems into Product Line Assets – *a Case Study*
Kyo Chul Kang, Moonzoo Kim, Jaejoon Lee, Byungkil Kim 45

Reuse without Compromising Performance: Industrial Experience from RPG Software Product Line for Mobile Devices
Weishan Zhang, Stan Jarzabek 57

Extracting and Evolving Mobile Games Product Lines
Vander Alves, Pedro Matos Jr., Leonardo Cole, Paulo Borba, Geber Ramalho 70

Short Papers

Determining the Variation Degree of Feature Models
Thomas von der Maßen, Horst Lichter 82

Modeling Architectural Value: Cash Flow, Time and Uncertainty
Jacco H. Wesselius 89

A Knowledge-Based Perspective for Preparing the Transition to a Software Product Line Approach
Gerardo Matturro, Andrés Silva 96

Strategies

Comparison of System Family Modeling Approaches
Øystein Haugen, Birger Møller-Pedersen, Jon Oldevik 102

Cost Estimation for Product Line Engineering Using COTS Components
Sana Ben Abdallah Ben Lamine, Lamia Labed Jilani, Henda Hajjami Ben Ghezala 113

Innovation Management for Product Line Engineering Organizations
Günter Böckle 124

Panels

Change is Good. You Go First
Moderator: Charles W. Krueger 135

A Competition of Software Product Line Economic Models
Moderator: Paul Clements 136

Validation

Enabling the Smooth Integration of Core Assets: Defining and Packaging Architectural Rules for a Family of Embedded Products
Tim Trew 137

Design Verification for Product Line Development
Tomoji Kishi, Natsuko Noda, Takuya Katayama 150

Scoping and Architecture

QFD-PPP: Product Line Portfolio Planning Using Quality Function Deployment
Andreas Helferich, Georg Herzwurm, Sixten Schockert 162

Product-Line Architecture: New Issues for Evaluation
Leire Etxeberria, Goiuria Sagardui 174

Strategies of Product Family Architecture Development
Eila Niemelä 186

Product Derivation

Defining Domain-Specific Modeling Languages to Automate Product Derivation: Collected Experiences
Juha-Pekka Tolvanen, Steven Kelly 198

Supporting Production Strategies as Refinements of the Production Process
Oscar Díaz, Salvador Trujillo, Felipe I. Anfurrutia 210

Using Variation Propagation for Model-Driven Management of a System Family
Patrick Tessier, Sébastien Gérard, François Terrier, Jean-Marc Geib 222

Author Index 235

Next Generation Software Product Line Engineering

David M. Weiss

Avaya Labs,
233 Mt. Airy Rd.,
Basking Ridge, NJ 07920
weiss@avaya.com

Software product line engineering has advanced to the point where we know how to create software product lines on small to medium scales, and some organizations are having success on a larger scale. Success has come rather slowly, however, if one considers that many of the key ideas are 25-35 years old. For example, Dijkstra discussed the idea of program families in the late 1960s, David Parnas and others clarified the idea and showed how to apply it in real-time systems in the mid 1970s, and Jim Neighbors invented domain analysis in the early 1980s. Through the 1980s and 1990s we saw the systematization of product line engineering processes and their first applications. The first Software Product Lines Conference was held in 2000. Much of the development of the field has focused on technical aspects of creating product lines and producing applications. Indeed, most of the technical problems in creating product lines now seem solvable for many product lines. The Software Product Line Hall of Fame gives us examples of successful large scale product lines.

Institutionalizing the use of product lines in industrial organizations on a large scale may now require overcoming the obstacles in creating the right organizations and in quantifying the economics. Institutionalization often founders on the question of whether to create an organizational unit dedicated to domain engineering and developing the product line engineering environment, or whether to distribute the domain engineering task among different organizational units. Are there other organizational choices that we can make that solve this problem? How do other industries, which cannot survive without creating product lines, solve this problem? The economic justifications are typically cast in terms of a simple, cost-based model. What, then, is a good model to use?

The questions for the next generation of product lines focus on the following.

1. What are reliable, repeatable techniques for creating large scale product lines and the organizations that produce them?
2. What is the right economic model for an organization to use in deciding what product lines to create?
3. What is the next step in bringing organization to the way that we think about product lines?

I will discuss some possible avenues of approach for each of these problems.

H. Obbink and K. Pohl (Eds.): SPLC 2005, LNCS 3714, p. 1, 2005.

Software Product Families in Nokia

Jan Bosch

Nokia Research Center,
Software and Application Technologies Laboratory,
Helsinki, Finland
Jan.Bosch@nokia.com

Abstract. The level of software development and maintenance investment in embedded products has increased considerably over the last decade. As software product families are providing a proven approach to managing the cost and quality of software artefacts, Nokia has exploited this approach to software development for many years. This paper presents some lessons learned and the key challenges for the successful use and evolution of software artefacts.

1 Introduction

Reuse of existing software artefacts can be viewed as the holy grail of software engineering. For close to four decades, we have, as a software engineering community, evolved through an extended set of techniques for achieving higher productivity, more dynamic, responsive software development and lower maintenance cost. Techniques proposed in this context include modules, components, libraries, object-orientation, frameworks, architecture and, of course, software product families.

Software product families can be viewed as addressing a specific area of software reuse as most published product families are of an embedded nature, combining mechanical, hardware and software elements and less focused on information systems style functionality. Although this division has long been an accurate one, there is a clear trend towards blurring the distinctions between these two categories of systems. Embedded systems are becoming increasingly networked, upgradeable after their initial deployment, able to dynamically embed in new contexts and record, process and store increasing amounts of data. Examples of these kinds of systems can, among others, be found in the telecom, consumer electronics and automotive industry.

The transition from traditional, closed embedded systems to a world in which embedded systems provide platforms for deploying a wide variety of distributed, possibly peer-to-peer applications has a number of implications for research in the area of software product families as well. These implications include the increasing importance of hierarchy in product families, the increased complexity of variability management, the balance domain and product engineering and the role of open-source software.

The goal and contribution of this article is an analysis of the aforementioned implications for research in software product families. This analysis is performed from the perspective of Nokia, but also includes experiences from other companies that I have

H. Obbink and K. Pohl (Eds.): SPLC 2005, LNCS 3714, pp. 2 – 6, 2005.

worked with in the past and from earlier research performed at the University of Groningen. Consequently, the results should be relevant for software engineering organizations in general.

The remainder of this article is organized as follows. In the next section, an overview of the three main software product families for mobile terminals at Nokia is presented. Subsequently, in section 3, a set of challenges is presented that companies, including to various extent Nokia, are concerned with. Finally, the paper is concluded in section 5.

2 Overview of Product Families at Nokia

Nokia is a 55.000 person Fortune-500 company with revenue of around 30 billion euros. The company is organized in four business groups, i.e. Networks, primarily selling telecom infrastructure equipment and associated services, and Mobile Phones, Multimedia and Enterprise Solutions, addressing different segments of mobile devices with products and associated services.

The mobile devices business groups employ three main platforms in their products, i.e. Series 40, Series 60 and Maemo, an open-source Linux-based platform. The platforms address, with some overlap, mobile devices with different feature sets and price points. However, these platforms also share some components, so there is hierarchy in the shared artefacts.

In terms of the maturity model that I presented in [1], the platforms organizations typically employ the highest maturity model, i.e. the configurable product base approach. This means that most new features required for products under development typically are first developed as part of the platform. Once the platform is released the product configures the new platform release for use and inclusion in the product functionality.

Series 40

The Series 40 platform is a closed, proprietary platform consisting of a in-house developed operating system, a cellular subsystem managing wireless, cellular connectivity and a subsystem managing the applications and interface to the user. The Series 40 platform is primarily intended for mobile phones with restricted extended functionality, but can be extended with applications written in Java.

Series 60

The Series 60 platform is an open platform based on the Symbian operating system. The platform is explicitly intended for 3rd party application developers who can develop applications using native C/C++, Java or a scripting language such as Python or Perl. The architecture consists of an adaptation layer between the hardware and the Symbian OS, the Symbian OS, the Series 60 layer and a layer containing the core applications and extended application suite.