The Third Software Product Line Conference

SPLC 2004

August 30 – September 2, 2004

Boston, Massachusetts

http://www.sei.cmu.edu/SPLC2004

Follow-up Information

- Jack Greenfield's keynote address (QuickTime movie)
- Pitfalls Panel Slides
- Proceedings of SPLC 2004
- 2004 Product Line Hall of Fame
  - Ericsson AXE Family of Telecommunications Switches
  - Ericsson AXE Family of Telecommunications Switches
  - Salion's Product Line of Revenue Acquisition Management Systems
  - Phillips' product line of software for high-end television sets
- Conference Photos
- Future of SPLC 2006 and PFE-6

SPLC 2004 Conference a Success!
The Software Engineering Institute's Third Software Product Line Conference, SPLC 2004, was an unprecedented success. The conference drew 200 attendees from across the globe with the majority of representatives coming from North America and Europe.

Paul Clements, SPLC 2004 chair and senior member of the technical staff at the SEI, stated that "we are here because we believe that software product line engineering is a powerful idea and that we have shared vision: product line development is a low-risk, high-return proposition for the entire software engineering community."

This year's conference offered 18 presentations of quality research and experience papers, 14 tutorials, 6 workshops, 3 panels, and 7 product line demonstrations. The event was highlighted by keynote speaker Jack Greenfield, Microsoft architect for enterprise and frameworks, who unveiled Microsoft's plans to support the concept of software factories.

Other highlights of the conference included four new inductions into the Software Product Line Hall of Fame. They were: General Motors' Powertrain product line, Salion, Inc.'s product line of revenue acquisition management systems, Ericsson's AXE family of telecommunication switches, and Phillips' product line of software for high-end television sets.

The Proceedings of SPLC 2004 are available from Springer as part of the Springer Lecture Notes in Computer Science series (LNCS, Vol. 3154).

Paul Clements	David Weiss	Rob van Ommering
Software Engineering Institute	*Avaya*	*Philips*
Conference Chair	*Program Cochair*	*Program Cochair*

Conference Information

- Keynote Address
- Program
- Papers
- Hall of Fame
- Tutorials
- Workshops
- Panels
- Demonstrations
- Program Committee
- About the Logo

The Software Engineering Institute (SEI) is proud to sponsor the third Software Product Line Conference (SPLC 2004).

The SEI is a federally funded research and development center sponsored by the U.S. Department of Defense and operated by Carnegie Mellon University.

The Third Software Product Line Conference

SPLC 2004

Panel Slides

Avoiding, Surviving, and Prevailing Over Pitfalls in Product Line Engineering
Panel Moderator: Charles W. Krueger, BigLever Software, Austin, TX, USA (slides)

Panelist
Jim Dager, Cummins, USA (slides)
Ulf Olsson, Ericsson, Celsius Tech (now Saab Tech) (slides)
Anders Heie, Nokia, USA (slides)
Martin Verlage, MARKET MAKER, Germany (slides)
Bill Hetrick , Engenio Information Technologies (formerly LSI Logic Storage Systems), USA (slides)

The Third Software Product Line Conference

SPLC 2004

2004 Product Line Hall of Fame

Four new software product lines were inducted into the Hall of Fame at SPLC 2004: General Motors' Powertrain product line, Salion, Inc.'s product line of revenue acquisition management systems, Ericsson's AXE family of telecommunication switches, and Phillips' product line of software for high-end television sets.

General Motors Powertrain (GMPT)

General Motors Powertrain (GMPT) is a division of General Motors Corporation (GM), the world's largest producer of passenger cars and trucks. GMPT provides powertrains to GM, its subsidiaries, and its partners. Today's automotive powertrains consist of an engine, transmissions and the associated control system. In the control system, there are electrical components, an electronic control module, and the software that runs this system. The software that runs this system is called the General Motors Powertrain Software Product Line.

GMPT began its transition to a product line approach for its embedded powertrain control software in the late 1990's. Since the inception of the product line, GMPT has taken four iterations of its software product line to production. During its short product line history, GMPT has used the software product line to support multiple electronic architectures - 5 engine control modules, 4 transmission control modules and 3 powertrain control module supporting both an engine and transmission controls. The software product line is also capable of supporting multiple powertrain physical architectures (diesel engines vs. gas engines, clutch-to-clutch transmissions vs. freewheel transmissions). Finally, controller products built using the GMPT software product line cleanly interface with over 100 vehicle platforms.

The GMPT software product line is now the basis for nearly all new control modules being developed within GMPT. In the next three years, GMPT expects to take the number of software sets supporting gasoline engine programs from 17 down to 3. In this manner, the use of the GMPT software product line has enabled GMPT to support the demand for new control modules and new powertrain features with minimal additional resources.

More information about the GMPT software product line is available at www.eecs.umich.edu/courses/eecs486/win03/notes/GMVisit.pdf

Ericsson AXE Family of Telecommunications Switches

The Ericsson AXE family of telecommunications switches was created in the 1970's as a conscious effort to create a system structure that would be resilient to a wide variety of changes:

- functionality requirements: telephony is constantly evolving as a services
- local adaptations: Ericsson is active in 140 countries, and many regional and national regulations must be accommodated
- scalability: the architecture must deliver cost-effective solutions from small installations to major

international hubs
- personnel: several generations of engineers have kept building on their predecessors' achievements, in dozens of development centers all over the world
- technology evolution, but more of that below!

In terms of technology growth, AXE has survived a remarkable set of changes. This goes from the first switch deployment in 1976, which actually controlled relay devices, through the introduction of digital switching, ATM, and now moving on to a softswitch architecture. In the latter case, AXE takes on the server role, controlling gateway devices based on various technologies including IP transport, carrying the AXE architecture into the Voice-over-IP world.

The software design of the AXE family was built from the ground up based on what we now describe as an object oriented approach. Objects (blocks) interact using message passing; simple and clear rules regarding design time and runtime semantics minimize the risk of design deterioration. The prime example of the resilience of the architecture is Ericsson's implementation of mobile switching on top of the AXE platform and basic switching application structure. This includes a number of analog cellular systems, as well as every major digital cellular standard: GSM/ GPRS/EDGE, WCDMA, D-AMPS, PDC, cdma2000, with others to follow.

During its 30+ year history, AXE has also evolved in terms of the computing platform, going through a number of performance enhancing technology changes. In the latest step, this has included successfully transferring the architecture onto a commercial hardware platform, while retaining software and interface compatibility with the wide range of line interfaces and switching devices available within the family.

Finally, AXE has been a considerable commercial success: several thousand switches are in operation all over the world, and 40% of all cellular phone calls are placed through Ericsson equipment.

Salion's Product Line of Revenue Acquisition Management Systems

In 2001, Salion was a startup preparing for its initial product launch. Even before completing its first product or getting its first customer, Salion knew it would have to solve the software product line problem – how to efficiently engineer a collection of nearly identical variations of their core product. The core product in this case was an enterprise software system to help optimize front-end processes for companies whose revenue stream starts by receiving and bidding on *requests for quotes* (RFQs).

Most of the product line methodologies being discussed at the time were *proactive*, requiring experience in a stable and well defined application domain. However, Salion expected some turbulence in their domain since they were inventing a new product in a new market and would be integrating the product into a wide variety of unknown customer settings.

At about the same time, BigLever Software was promoting the agile notion of *reactive* software product line engineering as well as lightweight and incremental adoption strategies using commercial off-the-shelf product line engineering and variation management technology. Salion decided to adopt this approach and in hindsight, this decision was a central part of their success in bringing their product line to market.

The unmodified architecture, design, and source modules from Salion's initial *baseline* product served as the core assets for the product line. By utilizing the existing baseline product for core assets and off-the-shelf technology for software product line engineering and variation management, Salion made the transition to a reactive software

product line approach in only 2 person-months of effort, which is two orders-of-magnitude less than previously reported efforts with proactive software product line transition efforts. They have subsequently deployed 16 commercial products in their product line, each with an average 95% software reuse and 90% reduction in engineering effort compared to the initial baseline product.

By taking a reactive software product line approach and using off-the-shelf product line engineering technology, Salion has been able to reap the benefits of software product lines while remaining agile in the face of turbulence in their application domain.

- Clements, P. and Northrop, L., *Salion, Inc.: A Software Product Line Case Study*, Software Engineering Institute (SEI) Technical Report CMU/SEI-2002-TR-038, Carnegie Mellon University, Pittsburgh, PA, November 2002.

- Buhrdorf, R. and Churchett, D. *Product Line Agility in the Face of Turbulence - The Salion Success Story*, on www.SoftwareProductLines.com.

- Krueger, C. and Churchett, D., *Eliciting Abstractions from a Software Product Line*, in Proceedings of the OOPSLA 2002 PLEES International Workshop on Product Line Engineering. Seattle, Washington. November 2002, pages 43-48.
- Buhrdorf, R., Churchett, D. and Krueger, C. *Salion's Experience with a Reactive Software Product Line Approach*, in Proceedings of the 5th International Workshop on Product Family Engineering. Siena, Italy. November 2003.

Philips Product Line of Software for Television Sets

Philips is one of the world's largest consumer electronics companies, and a global leader in televisions and other consumer products. While initially solely consisting of hardware, TVs now contain fully equipped embedded computers to control the hardware and to implement extra features. These computers started small, with 1 kilobyte of code around 1980, but software size has grown according to Moore's Law, resulting in many million lines of code today. While this by itself makes *complexity* an important issue, the other important issue is *diversity*, since televisions are produced in many different variants. To efficiently manage complexity and diversity, we decided to set-up a software product line.

The first step was the definition of a *software component model*, Koala, designed to allow the creation of different products by making new combinations of selections of reusable components. For this, all context dependencies of components are made explicit in the form of provides and requires interfaces, while diversity interfaces allow for internal variation of components. An explicit architectural description language, one of the first to be used extensively in industry, helps to manage complexity and to automatically generate product specific binding code. While Koala was modeled after Darwin and Microsoft COM, it is specifically tuned to build software product lines of resource-constrained products.

The second step was the creation of a *product line architecture*, profiting as much as possible from (re-engineered) existing software. This architecture is not a generic architecture shared by all products, but rather a set of rules and principles that should be obeyed by individual products, allowing each product to have its own specific and optimized architecture. The architecture was split in three main layers to anticipate on the use of 2nd and 3rd party software in the future.

To effectively deploy the architecture, changes had to be made to the existing *development processes*, which were optimized for the creation of single products. Typical changes were the introduction of component and interface data sheets to replace the traditional requirement specifications, and the definition of 'infinitely' progressing asset development projects, as opposed to finite product development projects.

The fourth step was the adaptation of the *development organization* to accommodate product line development. While we do separate asset from product development, all asset teams are aware of all products and vice versa, to ensure that the right level of generality is achieved. Another important issue was the alignment of the organization with the architecture, making sure that each subsystem is being developed on a single site.

The component model was created in 1996. The architecture was defined in 1998. The first product came on the market in 2000. Since 2002, all Philips' mid-range and high-end televisions have software derived from this product line. Today, there are 20 different software releases per year, where each release serving 1-5 different product types. The product line supports three different hardware platforms. When we started, diversity was one of the top three issues on the agenda of architects. Now, diversity has disappeared as issue entirely.

Our product line approach is extensively documented in http://irs.ub.rug.nl/ppn/275169561.

The Third Software Product Line Conference

SPLC 2004

Conference Photos













The Third Software Product Line Conference

SPLC 2004

Announcement

During the SPLC 2004 conference, the SPLC Steering Committee met and discussed the need to unite the software product line community worldwide to achieve greater impact. As a start, the committee decided beginning in 2006 to merge SPLC with its European counterpart, the Product Family Engineering (PFE) conferences, sponsored to date by the European Union. The plan is that the SPLC Steering Committee will lead three conferences, tentatively called SPLC-Europe (formerly PFE), SPLC-Americas, and SPLC-Asia. ESI will sponsor SPLC-Europe; the SEI will sponsor SPLC-Americas; and the Korea Software Institute will sponsor SPLC-Asia. The Steering Committee expanded its constituency to address its broader agenda.

SPLC 2004 ended with an announcement that the next SPLC-Americas, sponsored by the SEI, will be held in August 2006. John McGregor (Clemson University/SEI) will be the conference chair. Robert Nord (SEI) and Frank van der Linden (Philips Medical Systems) will be program co-chairs. PFE-6 will be held in France in 2005.

The SEI will establish an internet site, www.splc.net, to provide a focal point for capturing and announcing PFE and SPLC conferences.

The Third Software Product Line Conference

SPLC 2004

Technical Papers (Authors and Abstracts)

The following information provides the title, author(s), and abstract of the 18 technical papers accepted for SPLC 2004. The papers are listed in no particular order.

Governing Software Product Lines and Reorganizations

Truman Jolley, Boeing Commercial Airplanes
David Kasik, Boeing Commercial Airplanes
Tammy Ben, Boeing Commercial Airplanes

It's a fact of life that organizations love to reorganize. Reorganizations have a profound effect on the way product lines are governed. We introduce the concept of the Responsibility, Authority, and Accountability (RAA) network. An RAA network assists in the governance process of product lines for internal information systems, even in the face of massive reorganization. Armour ("Reorg Cycle") describes the pressures of reorganization to balance the "dimensions of organization" (e.g., geography, customers, product technology); we apply polarity management to balance the dimensions. Armour describes the difficulty of applying hierarchical organization charts–"single dimension management structures"–to the above "multidimensional environments"; we apply lean RAA networks to span organization charts and provide the multidimensional view needed for product lines. Armour observes that network organization approaches "do not have a good track record"; our experience is that lean, resilient RAA networks document the product line's governance architecture. These governance architecture patterns are applied repeatedly to the strata of products in the product line. We present the governance architect's RAA to define, monitor, and sustain governance health using these tools: polarity maps, polarity networks, RAA maps, and RAA networks.

Software Product Line Support in Coremetrics OA2004

James Snyder, Coremetrics, Inc.
Harry Lai, Coremetrics, Inc.
Shirish Reddy, Coremetrics, Inc.
Jimmy Wan, Coremetrics, Inc.

We present the transformation of the Coremetrics analytic application software engineering approaches from a traditional Java 2 Enterprise Edition (J2EE) environment to a software product line approach. Particular emphasis is placed on the motivation for a product line approach and the software engineering challenges faced particularly as an application service provider. We also discuss our definitions of product line variation and binding, and how we

provide tool support using the open source Integrated Development Environment Eclipse. We also provide our observations about the software product line approach, how our work and findings relate to other work in this area, and lastly, how configuration management plays an integral and complementary role in our approach.

Introducing Product Line Approach at Bosch Gasoline Systems: Experiences and Practices

Mirjam Steger, Robert Bosch GmbH
Christian Tischer, Robert Bosch GmbH
Wolfgang Stolz, Robert Bosch GmbH
Stefan Ferber, Robert Bosch GmbH

Software engineering in the automotive domain faces outstanding challenges in terms of quality, cost, and functional complexity. To ensure process and product excellence, Bosch Gasoline Systems (GS) introduced a process improvement program based on Capability Maturity Model Integration (CMMI) and adopted the product line approach (PLA). Business strategies for software products, software sharing with customers, and common solutions for diesel and gasoline engine control software are inputs for the product line architecture. The steps towards the PLA started with an evaluation project followed by an evolutionary rollout. The lack of suitable mechanisms and tools for some crucial nonfunctional requirements is a major drawback for introducing the PLA. Nevertheless, GS considers it the only systematic approach to dealing with current and future challenges.

Four Mechanisms for Adaptable Systems – A Meta-Level Approach to Building a Software Product Line

Claudia Fritsch, Robert Bosch GmbH
Burkhardt Renz, University of Applied Sciences Giessen-Friedberg

For more than ten years, we have developed and maintained a software product line of legal expert systems. They share certain functionality, such as interaction with the user by means of a graphical interface, capturing data, storing information in a database, and printing documents. They differ mainly in two areas: domain description and technical infrastructure.

When we designed the architecture for this software product line, we focused on two requirements in particular:

1. Domain experts should be involved in development, but should not have to learn a general-purpose programming language and
2. Changes in domain descriptions should leave technical code untouched–and vice versa.

Using a *meta-level architecture*, we achieved a sound decoupling: Domain descriptions are kept in the meta level. Appropriate engines included in the base level act according to these descriptions.

We present the four meta-level *mechanisms* that we have developed for the design of this software product line. They separate domain descriptions from technical code in the following areas:

- data reference and access;
- input and output control;
- application and domain logic; and
- user command control.

Automatic Generation of Program Families by Model Restrictions

Andrzej Wasowski, IT University of Copenhagen

We study the generative development of control programs for families of embedded devices. A software family is described by a single common model and restriction specifications for each of the family members. The model and the specifications are used in the automatic generation of restricted programs. We describe an application of the process of modeling reactive systems with statecharts. A trace inclusion refinement relation is established for automatically generated family members, inducing a behavioral inheritance hierarchy over the generated programs.

Software Product Lines in ArchJava

Sebastian Pavel, Ecole des Mines de Nantes
Jacques Noyé, Ecole des Mines de Nantes
Jean-Claude Royer, Ecole des Mines de Nantes

This paper considers the use of a state-of-the-art, general-purpose, component-programming language, specifically ArchJava, to implement software product lines. Component-programming languages provide a more straightforward mapping between components as assets and components as implementation artifacts. However, guaranteeing that the implementation conforms to the architecture raises new issues with respect to dynamic configuration. We show how this can be solved in ArchJava by making the components auto-configurable, which corresponds to replacing components by component generators. Such a scheme can be implemented in various ways, in particular with a two-stage generator. This solution goes beyond the initial technical ArchJava issue and complements the standard static generative approach to software product line implementation.

Software Product Family Evaluation

Frank van der Linden, Philips Medical Systems
Jan Bosch, University of Groningen
Erik Kamsties, University of Duisburg-Essen
Kari Känsälä, Nokia Research Center
Henk Obbink, Philips Research Laboratories

This paper proposes a four-dimensional evaluation framework for software product family engineering. The four dimensions relate to the software engineering concerns of business, architecture, organisation, and process. The

evaluation framework is intended for use within software developing organisa-tions to determine the status of their own software product family engineering and the priorities for improving. The results of the evaluation can be used for benchmarking, roadmapping, and developing improvement plans. An initial evaluation of a real industrial case is presented to show the validity of the framework.

Practical Evaluation of Software Product Family Architectures

Eila Niemelä, VTT Technical Research Centre of Finland
Mari Matinlassi, VTT Technical Research Centre of Finland
Anne Taulavuori, VTT Technical Research Centre of Finland

Faster time to market and decreased development and maintenance costs are goals most companies are trying to reach. Product family engineering (PFE) provides a means of achieving these goals. Product family architecture (PFA) is the key issue in family engineering. However, companies have to decide how to adopt PFE and how to develop their software PFA. This paper introduces the basic issues essential to PFA development, explores three different approaches to applying PFAs in industrial settings, and, finally, presents the evaluation results through an evaluation model of software product families.

On the Development of Software Product-Family Components

Jan Bosch, University of Groningen

Several approaches to the development of shared artefacts in software product families exist. Each has advantages and disadvantages, but there is no clear framework for selecting among these alternatives. As a consequence, mismatches between the optimal approach and the one currently used by an organization may lead to several problems, such as a high degree of erosion, mismatches between product needs and shared components, organizational "noise" and inefficient knowledge management. This paper (1) presents the problems resulting from the aforementioned mismatch, (2) presents the relevant decision dimensions that define the space of alternatives, (3) discusses the advantages and disadvantages of each alternative and (4) presents a framework for selecting the best alternative for each decision dimension based on a three-stage adoption model.

Experiences in Software Product Families: Problems and Issues during Product Derivation

Sybren Deelstra, University of Groningen
Marco Sinnema, University of Groningen
Jan Bosch, University of Groningen

A fundamental reason for investing in product families is to minimize the application engineering costs. Several organizations that employ product families, however, are becoming increasingly aware of the fact that, despite the efforts in domain engineering, deriving individual products from their shared software assets is a time- and effort-

consuming activity. In this paper, we present a collection of product derivation problems that we identified during a case study at two large and mature industrial organizations. These problems are attributed to the lack of methodological support for application engineering, and to underlying causes of complexity and implicit properties. For each problem, we provide a description and an example, while for each cause we present a description, consequences, solutions, and research issues. The discussions in this paper are relevant outside the context of the two companies, as the challenges they face arise in, for example, comparable or less mature organizations.

A Feature-Based Approach to Product Line Production Planning

Jaejoon Lee, Pohang University of Science and Technology
Kyo Kang, Pohang University of Science and Technology
Sajoong Kim, Korea IT Industry Promotion Agency

A production plan, which describes how core assets are used to develop products, has an important role in product line engineering as a communications medium between core asset developers and product developers. Recently, there have been efforts to address issues related to production planning, most of which focus on the process and business/management aspects of production planning; not much emphasis is given to technical issues such as deciding which features will be made as core assets and what their granularity will be. In this paper, we introduce a feature-based approach to product line production planning and illustrate how our approach addresses these technical issues. In our approach, a feature model and feature-binding information are used as primary input to production plan development. A product line production plan created using our approach could be customized easily to a product-specific production plan, because when we developed the approach, we considered units of product configurations as well as their integration techniques.

COVAMOF: A Framework for Modeling Variability in Software Product Families

Marco Sinnema, University of Groningen
Sybren Deelstra, University of Groningen
Jos Nijhuis, University of Groningen
Jan Bosch, University of Groningen

A key aspect of variability management in software product families is the explicit representation of the variability. Experiences at several industrial software development companies have shown that a software variability model should do four things: (1) uniformly represent variation points as first-class entities in all abstraction layers (ranging from features to code), (2) allow for the hierarchical organization of the variability, (3) allow for the first-class representation of simple (i.e., one-to-one) and complex (i.e., n-to-m) dependencies, and (4) allow for modeling the relations between dependencies. Existing variability modeling approaches support the first two requirements, but lack support for the latter two. The contribution of this paper is a framework for variability modeling–COVAMOF–that provides support for all four requirements.

Observations from the Recovery of a Software Product Family

Patricia Lago, Vrije Universiteit
Hans van Vliet, Vrije Universiteit

The problem of managing the evolution of complex and large software systems is well known. Evolution implies the reuse and modification of existing software artifacts, and this means that the related knowledge must be documented and maintained.

This paper focuses on the evolution of software product families, although the same principles apply in other software development environments as well. We describe our experience gained in a case study recovering a family of six software products. We give an overview of the case study, and provide lessons learned, implicit assumptions reconstructed during the case study, and some rules we think are generally applicable. Our experience indicates that organizing architectural knowledge is a difficult task. To properly serve the various uses of this knowledge, it needs to be organized along different dimensions and tools are required. Our experience also indicates that, next to variability explicitly designed into the product family, a ``variation creep" is caused by different, and evolving, technical and organizational environments of the products. We propose explicitly modeling invariabilities, next to variabilities, in software product lines to get a better grip on this variation creep.

Product Line Potential Analysis

Claudia Fritsch, Robert Bosch GmbH
Ralf Hahn, Robert Bosch GmbH

A *Product Line Potential Analysis* (PLPA)enables us to make a quick decision as to whether the product line approach (PLA) is suitable for a given set of products and target market. The PLA framework offers no support for this as yet.

A PLPA is executed in a half-day workshop. A structured interview based on a questionnaire examines products, software, markets, and customers. The answers are compared to a set of criteria for the applicability of the PLA. The analysis results primarily in the decisions: "yes" (the PLA is suitable for these products and markets), "no", or "investigation required". Up to now our team has performed four PLPAs.

We present the list of criteria, a part of our questionnaire, and the workshop format. We discuss the PLPA in the light of related work and its limits and lessons learned, and we look at future work.

Generalized Release Planning for Product-line Architectures

Louis Taborda, Macquarie Graduate School of Management

This paper elaborates on the coordination and management of evolving software product lines, where development teams work around a shared and reusable domain infrastructure. The trend away from monolithic applications and

towards component-based, product line architectures has enabled the de-velopment of complex software to be undertaken by autonomous and often, geographically separated teams. Delivering a complete product or product line requires significant coordination to bring the separate development streams to-gether, at agreed-upon points in the schedule, for integration and test. In such complex development scenarios, a Release Matrix has been proposed as a gen-eralization of release planning and tracking, addressing multiple products, com-ponents, and their interdependencies at an enterprise or marketplace level. Here, we describe the results of the practical trials of the Release Matrix that provide pragmatic guidelines for its use and indicate areas for future research. Relation-ships to established processes, including requirements engineering and configu-ration management, are clarified, and the methodology-neutral technique is shown to complement work in areas, including Agile Methods and component contracts.

A Methodology for the Derivation and Verification of Use Cases for Product Lines

Alessandro Fantechi, University of Florence
Stefania Gnesi, I.S.T.I. - C.N.R.
Giuseppe Lami, I.S.T.I. - C.N.R.
Emiliano Nesti, University of Florence

In this paper, we present a methodology to express, in a formal way, the requirements of products belonging to a product line. We relied on a formalism allowing the representation of variabilities at the family level and the instantiation of them in order to move to the requirements of a single product. The proposed methodology also allows the formalization of the family constraints to be taken into account for the construction of the products belonging to it, along with the verification of the compliance to those constraints of a single product requirements document. This approach is promising due to its simplicity and effectiveness for being supported by automatic tools.

Staged Configuration Using Feature Models

Krzysztof Czarnecki, University of Waterloo
Simon Helsen, University of Waterloo
Ulrich Eisenecker, University of Applied Sciences Kaiserslautern

Feature modeling is an important approach to capturing commonalities and variabilities in system families and product lines. In this paper, we propose a cardinality-based notation for feature modeling, which integrates a number of existing extensions of previous approaches. We then introduce and motivate the novel concept of staged configuration. Staged configuration can be achieved by the stepwise specialization of feature models. This is important because in a realistic development process, different groups and different people eliminate product variability in different stages. We also indicate how cardinality-based feature models and their specialization can be given a precise formal semantics.

Scenario-Based Decision Making for Architectural Variability in Product Families

Pierre America, Philips Research
Dieter Hammer, Technical University Eindhoven
Mugurel Ionita, Technical University Eindhoven
Henk Obbink, Philips Research
Eelco Rommes, Philips Research

In this paper, we present a systematic approach towards decision making for variability in product families in the context of uncertainty. Our approach consists of the following ingredients: a suitable set of architectural views that bridge the gap between customer needs and available technology, a multi-view variation modeling technique, the selection of several scenarios of different kinds, and a quantitative analysis of quality aspects for these scenarios.

The Third Software Product Line Conference

SPLC 2004

Conference Tutorials

John D. McGregor
Clemson University
Tutorials Chair

The SPLC2004 will host fourteen conference tutorials:

- Using Domain-Specific Languages, Patterns, Frameworks, and Tools to Assemble Applications – Jack Greenfield
- Product Line Analysis – Gary Chastek and Patrick Donohoe
- Industrial-Strength Software Product line Engineering – John Klein, Deborah Hill, and David Weiss
- Quality Assurance for Software Product Lines – Ronny Kolb and Dirk Muthig
- Product Line Architectures for Global Software Development – Daniel J. Paulish, Roman Pichler, and Wolfgang Kuhn
- Architecture-Centric Software Engineering – Jan Bosch
- Software Variability Management – Jan Bosch
- Designing Software Product Lines with the Unified Modeling Language (UML) – Hassan Gomaa
- Developing a Measurement Program for Software Product Lines – Sholom Cohen, Dave Zubrow, and Gary Chastek
- Starting Product Lines (I) — Systematic Product Line Planning and Adoption – Klaus Schmid and Isabel John
- Starting Product Lines (II) — Product Line Analysis and Modeling – Isabel John and Klaus Schmid
- Generative Software Development – Krzysztof Czarnecki
- An Introduction to Software Product Lines – Linda M. Northrop and Paul C. Clements
- Adopting Software Product Lines – Linda M. Northrop and Lawrence Jones

---

Using Domain-Specific Languages, Patterns, Frameworks, and Tools to Assemble Applications
Jack Greenfield, Microsoft Corporation, Redmond, WA.

Increasingly complex and rapidly changing requirements and technologies are making application development increasingly difficult. This half-day tutorial explores this phenomenon and presents the Software Factory pattern for building languages, patterns, frameworks, and tools for specific domains, such as user interface construction or database design. We then explore innovations, such as adaptive assembly, software product lines, and model-driven development, that reduce the cost of implementing the pattern, making it cost-effective for narrower and more specialized domains, such as B2C application development and business process automation. We introduce the concept of the software schema, a network of viewpoints describing artifacts comprising the members of a family of software products, and we show how mappings between these viewpoints can be used to provide constraints

supporting model transformation and self-organizing processes. Finally, we discuss the formation of software supply chains and show how the Software Factory pattern distributes across organizational boundaries.

This tutorial is based on the book *Software Factories: Assembling Applications with Languages, Patterns, Frameworks and Tools*, by Jack Greenfield and Keith Short, to be published by John Wiley and Sons in July 2004.

Tutorial Objective
Participants will learn about the methods, practices, and technologies used in the development of software factories, including commonality and variability analysis, concrete and abstract syntax definition, model validation, model transformation, and software product line engineering.

Product Line Analysis
Gary Chastek and Patrick Donohoe, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.

Product line analysis (PLA) is early requirements engineering for a product line of software-intensive systems. It encompasses the rapid elicitation, analysis, specification, and verification of the requirements for a product line. The premise of product line analysis is that a sound initial understanding of the problem to be solved is essential before an organization embarks on a software product line as a solution.

This tutorial provides practitioners with a practical introduction to product line requirements modeling. It describes the semantics, properties, and interrelationships of the PLA work products that constitute a model of the product line requirements and provides guidance on their construction and use. A running example, based on home automation systems, illustrates PLA concepts.

The goal of the tutorial is to provide attendees with an understanding of how to capture and represent product line requirements in a systematic, repeatable way, and in a form that can be easily communicated to product line architects and product builders. The intended audience is practitioners (i.e., the product line requirements engineers), product line architects, and technology developers interested in creating supporting tools. Attendees are assumed to have some familiarity with software product lines and software analysis. Familiarity with object technology, including object modeling and use cases, is also assumed.

This is a half-day introduction to product line analysis. It provides a quick overview rather than a detailed description. By completing this tutorial, attendees should be able to

- define product line analysis
- explain its purpose and benefits
- apply the concepts to their own software product lines
- understand the importance of tailoring requirements for the product line architect

---

Industrial-Strength Software Product line Engineering
John Klein, Deborah Hill, and David Weiss, Avaya Labs, Basking Ridge, NJ.

Software product line engineering is one of the few approaches to software engineering that shows promise of improving software productivity by factors of 5 to 10. But, there are still relatively few examples of its successful application on a large-scale project, due in part to the complexity of the problem of initiating a product line engineering project and to the many factors that must be addressed for such a project to be successful. Practitioners and researchers alike are confronted with issues such as the following:

- What approaches have been tried? What is successful?
- How do I construct a convincing business case for product line engineering?
- What are the key activities needed to start a product line engineering project and to have it be successful? Can an incremental approach be used?
- How do I measure the effectiveness of product line engineering in my organization?
- How do I maintain momentum during the transition to product line engineering?

Tutorial Objectives

This tutorial draws on experience in introducing and sustaining product line engineering in Lucent Technologies and Avaya to answer such questions. Participants will take away an understanding of what drives a large software development organization to use product line engineering, the obstacles that can be encountered in putting product line engineering theory into practice, and practical approaches to overcoming such obstacles in a systematic way. Participants will learn both technical and organizational aspects of the problem. For example, the tutorial discusses areas that are critical to success in product line engineering but typically not covered by theory, such as who must be convinced to use product line engineering and what arguments to use. Finally, participants should leave the tutorial with ideas on how to transition a large organization to product line engineering while it is in the midst of developing and delivering products.

Participants will receive an inside look at an organization that is transitioning into product line engineering on a large scale. Success in such a project depends on both technical issues in creating and evolving a product line, and organizational issues in transforming an organization to be compatible with the product line engineering process.

Tutorial Approach

The tutorial will be conducted as a series of lecture and interactive sessions describing

- the process for creating a software product line
- creating an organization suitable for sustaining a software product line
- the time line of events that are needed to transition an organization from a single-system development approach to a product line approach

The tutorial covers both theory and practice, showing how the theory is converted into the practice, how the theory guides the practice, and how the theory is refined as a result of the practice. The theory comes from the Family-oriented Abstraction, Specification, and Translation (FAST) process, a trial of which was first conducted on a small scale in Lucent Technologies Bell Laboratories. The process is now being applied on a large scale in Avaya Laboratories. The project has faced all of the typical pressures seen in software development organizations today: pressure to reduce costs, to meet time-to-market deadlines, to improve quality, to introduce a more efficient software development process without disrupting current development projects, and to move into new markets that introduce greater variability in the product line.

Quality Assurance for Software Product Lines
Ronny Kolb and Dirk Muthig, Fraunhofer Institute for Experimental Software Engineering (IESE), Kaiserslautern, Germany.

The product line approach to software development is based on the systematic, large-scale reuse of development artifacts such as architecture, design, and components between a set of functional similar products. It promises, among other things, to shorten the development time of software systems and to significantly reduce development and maintenance costs. To achieve the promised improvements, however, the components and artifacts intended for reuse must be of high quality. Therefore, more than for traditional software development, quality assurance becomes a very crucial part of every product line effort.

Even though quality assurance has become not only more critical for software product lines, but also more complex due to the special properties of product lines such as genericity of software components, research in the field of software product lines to date has focused primarily on analysis, design, and implementation. In particular, the quality assurance challenges that arise in a product line context have been addressed insufficiently so far, and there is little guidance for product line organizations on how to systematically assure the quality of their product lines and reusable artifacts.

The overall goal of the tutorial is to provide an understanding of the problem of quality assurance in the context of software product lines and its importance for successful product line development. In particular, the tutorial aims at providing attendees with a detailed understanding of how the quality assurance process for product lines and generic software components needs to be different from traditional software systems and how quality assurance can be performed in the context of software product lines. To this end, the tutorial provides a discussion of the difficulties and challenges of quality assurance for software product lines and investigates the implications of product lines and reusable components on quality assurance. Finally, the tutorial aims to provide an understanding of existing quality assurance techniques and methods and how they can be applied in a product line context.

The tutorial addresses industrial practitioners, as well as applied researchers, working in the area of quality assurance or software product lines. In particular, the tutorial provides researchers with a concise overview of the current state of the art of quality assurance in the context of software product lines. In addition, the tutorial presents relevant issues and approaches in the area of quality assurance for software product lines, while providing industrial practitioners with a profound understanding of best practices they can apply.

Product Line Architectures for Global Software Development
Daniel J. Paulish, Roman Pichler, and Wolfgang Kuhn, Siemens Corporate Research, Inc., Princeton, NJ.

This half-day tutorial discusses how product line practices can be applied to software projects with development teams distributed around the world. Software products are growing in complexity, and the development organizations that implement new features are growing in staff size. Business managers are seeking new approaches, such as offshoring and outsourcing, to get new software products to market quicker, while reducing their overall development investments. Siemens has been performing research to decompose large-scale requirements into a well-structured set of software components that can be developed in parallel among globally distributed development teams. This tutorial describes an approach using product line architecture methods for multi-site development

projects in which software components are commissioned by a central product management and engineering organization for development at distributed sites.

As a participant in this tutorial, you will learn product line approaches based on best practices for developing your software products using globally distributed development organizations. Our approach for global development consists of improved practices in three key areas:

1. requirements engineering: Model the functionality using current best practices and drive all aspects of the product line solution development from the model.
2. software architecture: Drive the product line towards standard common data models and a component framework that will help enable integration.
3. global development: Optimize the organization of product solution development using small, agile distributed component development teams synchronized by a central organization.

A central product management and engineering team controls the requirements model and high-level product architecture. The requirements model and architecture are designed such that component sizes are defined to be relatively small with a maximum specified size in terms of lines of code (LOC), function points, development time, and development effort. The distributed development teams are constrained with respect to functionality, delivery schedule, effort, and schedule for developing their commissioned components. However, they are free to use any local agile processes as long as they meet the constraints. Central product management and engineering synchronize the concurrent development of the planned components and their functionality and are responsible for component acceptance testing and integration.

This tutorial is oriented towards experienced software engineers, architects, and project and technical managers who are involved with planning or executing distributed development projects. The goal of the tutorial will be to communicate product line best practices that will help minimize the risks associated with global software development.

Architecture-Centric Software Engineering
Jan Bosch, University of Groningen, Department of Computing Science, The Netherlands.

Many software organizations are in the process of moving from project- and product-centric software engineering to architecture-centric software engineering. Typically, this move is made for two reasons: (1) the architecture allows for a clear breakdown in parts, whereas a project-centric approach easily leads to a monolithic system and (2) the organization is interested in exploiting the commonalities between its products or systems.

This tutorial addresses this development by providing an overview and in-depth treatment of the issues around architecture-centric software engineering [1]. The first topic is concerned with the design of software architectures in the presence of existing components and infrastructure (e.g., designing the architecture in a top-down or bottom-up fashion). Two issues discussed in the context of architecture design are the notion of architecture design decisions and software variability management. Architecture design decisions lack first-class representation in the architecture descriptions. We are developing solutions to address this. Software variability management is concerned with explicitly managing the points of variation in software artifacts and in the software architecture in particular.

The second topic is the evaluation and assessment of software architectures. As architectural changes late in a

development project or during evolution are often prohibitively expensive, verifying that the architecture has the right quality properties is of great importance. The aim is to, preferably quantitatively, predict the qualities of a software system based on its software architecture.

The final topic of the tutorial is concerned with the use of the software architecture, especially in the context of software product families and highly configurable products. In this part, not only the technical aspects, but also the process and organizational viewpoints are discussed, and the relation between the different dimensions is presented. In addition, evaluation models and adoption approaches are discussed. The topics are illustrated extensively by examples and experiences from many industrial cases.

Reference

1. Jan Bosch, Design and Use of Software Architectures: Adopting and Evolving a Product Line Approach, Pearson Education (Addison-Wesley & ACM Press), ISBN 0-201-67494-7, May 2000.

Software Variability Management

Jan Bosch, University of Groningen, Department of Computing Science, The Netherlands.

In a variety of approaches to software development, software artefacts are used in multiple contexts or for various purposes. The differences lead to so-called variation points in the software artefact. During recent years, the amount of variability supported by a software artefact is growing considerably, and its management is developing as a main challenge in the development, usage, and evolution of software artefacts. Areas where the management of variability is evolving as a challenge include software product families [1], component-based software development, object-oriented frameworks, and configurable software products such as planning systems for enterprise resources. For example, in a typical software product family, the number of variation points may easily range in the thousands.

Software variability is the ability of a software system or artefact to be changed, customized, or configured for use in a particular context [2]. A high degree of variability allows the use of software in a broader range of contexts (i.e., the software is more reusable). Variability can be viewed as consisting of two dimensions: space and time. The space dimension is concerned with the use of software in multiple contexts (e.g., multiple products in a software product family). The time dimension is concerned with the ability of software to support evolution and changing requirements in the software's various contexts.

Successful management of variability in software artefacts leads to better customizable software products that are, in turn, likely to result in higher market success. In the information systems domain, the products are more easily adaptable to the needs of different user groups; in the embedded systems domain, the software can be more easily configured to work with different hardware and environmental constraints.

The tutorial first establishes the importance of software variability management, among other things through industrial examples consisting of thousands of variation points and dependencies. Second, the tutorial defines the concept of variability and discusses notational and visualization aspects, including the COVAMOF model. Third, we discuss the assessment of software artefacts for variability (i.e., COSVAM) and the design of architectures and components for variability. Fourth, the use of variation points is presented (e.g., while configuring instantiated software artefacts). Finally, some advanced issues including variation versus composition are discussed.

References

1. Jan Bosch, Design and Use of Software Architectures: Adopting and Evolving a Product Line Approach, Pearson Education (Addison-Wesley & ACM Press), ISBN 0-201-67494-7, May 2000.
2. Jilles van Gurp, Jan Bosch, Mikael Svahnberg, On the Notion of Variability in Software Product Lines, Proceedings of The Working IEEE/IFIP Conference on Software Architecture (WICSA 2001), pp. 45-55, August 2001.

Designing Software Product Lines with the Unified Modeling Language (UML)

Dr. Hassan Gomaa, Department of Information and Software Engineering, George Mason University, Fairfax, Virginia.

This tutorial addresses how to develop object-oriented requirements, analysis, and design models of software product lines using the Unified Modeling Language (UML) 2.0 notation. During requirements modeling, the tutorial covers how to develop kernel, optional, and alternative use cases for defining the software functional requirements of the system. The tutorial also describes the feature model for capturing product line requirements and how it relates to the use case model. During analysis, the tutorial covers how to develop static models for defining kernel, optional, and variant classes and their relationships. It also describes how to create dynamic models in which interaction models describe the dynamic interaction between the objects that participate in each kernel, optional, and alternative use case, and in which statecharts define the state-dependent aspects of the product line. The tutorial then covers how to develop component-based software architecture for the product line using the new UML 2.0 notation for structured classes and composite structure diagrams. That notation allows components, ports, and connectors, as well as provided and required interfaces, to be depicted. The tutorial gives an overview of the architectural structure patterns and architectural communication patterns that can be used in designing component-based product lines. The tutorial is illustrated by several examples and based on the book by Hassan Gomaa titled *Designing Software Product Lines with UML* to be published by Addison-Wesley in July 2004.

Developing a Measurement Program for Software Product Lines

Sholom Cohen, Dave Zubrow, and Gary Chastek, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.

Product line management should use measurement to anticipate the future rather than to simply record the past. The benefit and value of software measurement for product lines comes from the decisions and actions taken to support the product line in response to analysis of the data, not from the collection of the data. Addressing this management need is the primary goal of this tutorial.

Implementing measures for a software product line requires coordination across multiple projects. They must establish common goals and develop measures to track results on individual projects. We have developed this tutorial to help product line managers, software product development managers, software core asset development managers, and Software Engineering Process Group (SEPG) members support that activity. The tutorial will help the audience set measurable goals and determine if the software product line is producing the expected results. To benefit from the tutorial, attendees need no prior experience in measurement.

While the half-day format does not provide hands-on interaction, it will describe three sets of activities to develop a measurement program for software product lines:

1. Identifying Goals – explains transitioning from high-level product line goal statements to actionable measurement goals
2. Defining Indicators – describes the charts, tables, or measures that will address the software product line manager's goals
3. Creating an Action Plan – shows a plan for implementing the defined indicators

By following this approach, the risk of having data gathered, but not used, is minimized.

The tutorial also describes a range of measurements relevant to software product lines. The measures suggested here range from relatively mature to those whose general utility has yet to be validated. Therefore, a product line organization needs to assess its ability to generate the measures and the value those measures are likely to return to the organization. In most cases, an organization may wish to start with a subset of the product line measures described.

A case study in establishing a software product line measurement program illustrates the approach and use of software product line measurements. The organization covered in the case study established a Software Measurement Team to develop and monitor a measurement program. That measurement team includes representatives from programs using a core asset base. Four projects currently contribute to the measurement program.

Starting Product Lines (I) — Systematic Product Line Planning and Adoption
Klaus Schmid and Isabel John, Fraunhofer IESE, Sauerwiesen 6, 67661 Kaiserslautern, Germany.

To successfully and effectively adopt a software product line approach, the transition must be well aligned to the specific product line situation. As more and more organizations aim at a product line transition, this becomes increasingly an issue. To successfully and effectively adopt a software product line approach, a thorough analysis of the economic implications of the adoption must be performed, and the product line introduction needs to be adequately planned. This requires a precise picture of the product line through product analysis and modeling. A thorough analysis of the economic implications of the adoption must be performed, and the introduction of the product line needs to be correspondingly planned. Of course, such a transition has serious ramifications for the component structure of the software. We will discuss these consequences and show how the economic analysis itself can be used as a basis for deriving an adequate structure for the software. Thus, this tutorial provides a concise overview of the current state of the art of product line planning and adoption that is aimed at both researchers and practitioners of product line development.

Structure
The tutorial covers the following topics:

- introduction to product line development
- product line economics as a basis of software product line adoption and planning
- product line planning: aligning the product line plan with the business strategies of the organization
- overview on product line scoping: This includes an overview of existing scoping technologies and their

advantages and disadvantages. We will also discuss the three levels of scoping: (1) product portfolio scoping, (2) domain scoping, and (3) asset scoping.
- product line adoption and transition to product line development
- product line management
- product line evolution covers analyzing the impact of new products, introducing the knowledge about the existing asset base, and steering the ongoing development from an economic point of view.

Starting Product Lines (II) — Product Line Analysis and Modeling
Isabel John and Klaus Schmid, Fraunhofer IESE, Sauerwiesen 6, 67661 Kaiserslautern, Germany.

Product line engineering is recognized as a viable approach to large-scale software reuse. This tutorial provides a concise overview of the current state of the art of product line analysis and modeling and aims at giving an understanding of how to identify, analyze, and model commonalities and variabilities. It also provides an overview of the vast range of existing techniques for product line analysis and modeling. In particular, this tutorial provides researchers with a better understanding of the breadth of relevant issues and approaches, while providing industrial practitioners with a profound understanding of best practices they can apply. As the systematic identification and description of commonalities and variabilities are key in product line development to achieving successful reuse, the adequate selection or extension of modeling techniques can be regarded as a key to the overall goal of the SPLC. The modeling tutorial is a half-day tutorial, however, it can be combined with the tutorial titled "Starting Software Product Lines (I) - Systematic Product Line Planning and Adoption" to yield a full-day tutorial covering the early phases of product line development.

Structure

The tutorial covers the following topics:

- the importance of product line analysis and modeling as a key factor for successful product line engineering
- key principles of product line analysis and modeling (e.g., commonality and variability, decision modeling, domain analysis, application analysis, and traceability to all interrelated phases)
- overview of product line analysis and modeling techniques. We will also show how existing system modeling techniques (like e.g., UML use cases) can be adapted for modeling product line requirements.
- application analysis and derivation of product-specific models
- the PuLSE-CDA approach to product line modeling and analysis as an example modeling approach
- the interaction of analysis and modeling with other product line engineering phases (scoping and architecture and application modeling)

Generative Software Development
Krzysztof Czarnecki, University of Waterloo, Canada.

Abstract
System family engineering seeks to exploit the commonalities among systems from a given problem domain while managing the variabilities among them in a systematic way. In system family engineering, new system variants can be created rapidly based on a set of reusable assets (such as a common architecture, components, models, etc.) [1].

Generative software development aims at modeling and implementing system families in such a way that a given system can be automatically generated from a specification written in one or more textual or graphical domain-specific languages [2–8]. In this tutorial, participants will learn how to perform domain analysis (i.e., capturing the commonalities and variabilities within a system family in a software schema using feature modeling), domain design (i.e., developing a common architecture for a system family), and implementing software generators using multiple technologies, such as template-based code generation and model transformations. The relationship to model-driven development will be also discussed. The presented concepts and methods will be demonstrated using a case study.

References

1. Clements, P., Northrop, L.M.: Software Product Lines: Practices and Patterns. Addison-Wesley (2001)
2. Neighbors, J.M.: Software Construction using Components. PhD thesis, Department of Information and Computer Science, University of California, Irvine (1980) Technical Report UCI-ICS-TR160; available at http://www.bayfronttechnologies.com/ thesis.htm.
3. Cleaveland, J.C.: Building application generators. IEEE Software 5 (1988) 25–33
4. Weiss, D.M., Lai, C.T.R.: Software Product-Line Engineering: A Family-Based Software Development Process. Addison-Wesley (1999)
5. Czarnecki, K., Eisenecker, U.W.: Generative Programming: Methods, Tools, and Applications. Addison-Wesley (2000)
6. Cleaveland, C.: Program Generators with XML and Java. Prentice-Hall (2001)
7. Batory, D., Johnson, C., MacDonald, B., von Heeder, D.: Achieving extensibility through product-lines and domain-specific languages: A case study". ACM Transactions on Software Engineering and Methodology (TOSEM) 11 (2002) 191–214
8. Greenfield, J., Short, K.: Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools. Wiley (2004) To be published.

An Introduction to Software Product Lines

Paul Clements and Linda Northrop, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA

Software product lines have emerged as a new software development paradigm of great importance. A software product line is a set of software-intensive systems sharing a common, managed set of features, and that are developed in a disciplined fashion using a common set of core assets. Organizations developing their family of products as a software product line are experiencing order-of-magnitude improvements in cost, time to market, staff productivity, and quality of the deployed products.

This tutorial provides an overview of software product lines. It covers the basic concepts of software product lines, the essential software engineering and management practices, and a sampling of product line practice patterns that help organizations apply the practices in a way best suited to their individual needs. The concepts are illustrated with a detailed case study of an actual organization's experiences with the software product line approach. This tutorial is based on the book *Software Product Lines: Practices and Patterns* by Paul Clements and Linda Northrop.

This tutorial is appropriate for managers and practitioners. Participants should have experience in designing and developing software-intensive systems and some familiarity with modern software engineering concepts and management practices. The goal of the tutorial is for participants to understand the essential activities and practices involved in a software product line approach and to appreciate software product lines as an effective reuse strategy.

Adopting Software Product Lines

Linda Northrop and Lawrence Jones, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA

The tremendous benefits of taking a software product line approach are well documented. Organizations have achieved significant reductions in cost and time to market and, at the same time, increased the quality of families of their software systems. However, to date, there are considerable barriers to organizational adoption of product line practices. Phased adoption is attractive as a risk reduction and fiscally viable proposition.

This tutorial describes a phased, pattern-based approach to software product line adoption. It begins with a discussion of software product line adoption issues and then presents the *Adoption Factory*, a variant of the *Factory* pattern. The *Factory* pattern describes the entire product line organization. The *Adoption Factory* pattern provides a roadmap for phased, product line adoption. The tutorial covers the *Adoption Factory* in detail, including focus areas, phases, subpatterns, related practice areas, outputs, and roles. Examples of product line adoption plans following the pattern are used to illustrate its utility. The tutorial also describes strategies for creating synergy within an organization between product line adoption and ongoing CMMI or other process improvement initiatives.

The objective of the tutorial is to acquaint participants with product line adoption barriers and two ways to overcome them:

1. a phased, pattern-based adoption approach
2. explicit linkage with other improvement efforts

Participants should have experience in designing and developing software-intensive systems and familiarity with software product line concepts.

The Third Software Product Line Conference

SPLC 2004

Conference Workshops

Important: Workshop Attendance Criteria

Klaus Schmid
Fraunhofer IESE
Workshop Chair

SPLC 2004 will host six conference workshops:

- SPLiT - Workshop on Software Product Line Testing
- SPLYR - The First Software Product Lines Young Researchers Workshop
- Modeling Business Issues of Software Product Lines
- Quality Assurance in Reuse Contexts
- Software Variability Management for Product Derivation - Towards Tool Support
- Solutions for Automotive Software Architectures: Open Standards, References, and Product Line Architectures

Submission deadline is May 30th unless otherwise noted.

SPLiT - Workshop on Software Product Line Testing
Birgit Geppert, Avaya Labs, Basking Ridge, NJ, USA
Charles Krueger, BigLever Software, Austin, TX, USA
J. Jenny Li, Avaya Labs, Basking Ridge, NJ, USA

Workshop code: WS0439

Product line engineering (PLE) has become a major topic in industrial software development, and many organizations have started to consider PLE as state of the practice. One topic that needs greater emphasis is testing of product lines. Product line testing is crucial to the successful establishment of PLE technology in an organization. The workshop aims at addressing some of the open fundamental challenges of testing in a PLE setting. How can we manage the complexity of the test space? Can we leverage our established testing tools and procedures? A particularly hard challenge for test groups in a PLE setting is keeping pace with development productivity gains. If software developers can create unique product instances 10 times faster using PLE techniques, how does the test organization keep pace without having to hire 10 times as many test engineers? Are there PLE techniques that can provide efficiency gains for testing similar to those for development? These are questions that we have to face when transitioning to PLE, and, without adequate answers, testing becomes the bottleneck in PLE.

In this workshop, we aim at bringing together both researchers and practitioners from testing and PLE on all aspects of product line testing, from designing test cases with variation points over test coverage to testing tools. We are especially interested in exchanging industrial experience in product line testing and comparing different approaches to enable an integration of different ideas. Our goal is to provide a context for such an information exchange and to provide an opportunity to discuss innovative ideas, set a research agenda, and start collaborations on this topic. To achieve this, we invited experts not only from PLE, but also from testing, to participate in organizing the workshop and to help bring together both worlds and make this effort a success. Our program committee consists of the following members:

For PLE:

- Guenter Boeckle, Siemens AG, Germany
- Jan Bosch, University of Groningen, The Netherlands
- Krzysztof Czarnecki, University of Waterloo, Canada
- John McGregor, Clemson University, USA
- Dirk Muthig, Fraunhofer IESE, Germany
- Frank Roessler, Avaya Labs, USA

For Testing:

- Hira Agrawal, Telcordia, USA
- John Linn, Texas Instruments, USA
- Henry Muccini, University of L'Aquila, Italy
- Mladen A. Vouk, North Carolina State University, USA
- Eric Wong, University of Texas, USA

The results of the workshop will be published as part of the Avaya Labs report series and will be available online at http://www.research.avayalabs.com/. More information about the workshop is provided on the workshop's homepage http://www.biglever.com/split2004/.

SPLYR - The First Software Product Lines Young Researchers Workshop

Birgit Geppert, Avaya Labs, Basking Ridge, NJ, USA
Isabel John, Fraunhofer IESE, Sauerwiesen 6, 67661 Kaiserslautern, Germany
Giuseppe Lami, ISTI CNR, via Moruzzi, 1; 56124 Pisa, Italy

Workshop code: WS0427

The Software Product Lines Young Researchers (SPLYR) workshop addresses research activities in the field of software product lines (SPLs). Topics of interest include all aspects of developing, managing, evaluating, reusing, and maintaining SPLs. The peculiarity of this workshop is that it is specifically addressed to young researchers having original ideas and initiatives in the SPL field. We address mainly PhD work in progress but also encourage the submission of other work in progress such as master's or diploma theses. Another characteristic of this workshop is that it will not involve blind reviews. Instead, each young researcher will be assigned one panelist/reviewer whose name will be disclosed as part of the review report. This provides a unique opportunity for the participating young researchers to get in contact with their reviewers and to receive valuable input for their work and presentations before the actual workshop takes place.

The workshop itself aims at providing a platform for young researchers to present their work to an international audience and to discuss it with their peers and experts in the field. The panelists will comment on the presentations and give feedback for further developing the work. This represents a unique opportunity for the presenting young researchers to receive invaluable feedback from the panelists, to get in contact with other researchers in the field, to

present their work professionally, and to become familiar with other approaches and future research topics.

Panelists
- Len Bass, Software Engineering Institute, USA
- Jan Bosch, University of Groningen, Netherlands
- André van der Hoek, University of California, Irvine, USA
- Dirk Muthig, Fraunhofer IESE, Germany

Submissions to SPLYR should be sent to SPLYR@isti.cnr.it by May, 30th.

Modeling Business Issues of Software Product Lines
Sholom Cohen, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA

Workshop code: WS0437

Many organizations require financial justification before proceeding with a product line approach. The approach may appear very attractive in an intuitive sense, and it offers the obvious benefits of a faster time to market and higher quality. But, without the cost figures, the decision makers won't budget funds or personnel resources to carry out the up-front asset construction tasks. In addition, not all organizations are ready to commit up front to a full asset set—one that covers most if not all product line features. They favor a more incremental approach that tackles the areas of highest and most readily available commonality first.

Business modeling is a fundamental practice that provides input into a number of decisions that are made by organizations using or considering using the product line strategy. The purpose of this workshop is to present and discuss models that support the estimation of the costs and benefits in a product line development organization. The models should support decisions such as whether to use a product line strategy in a specific situation and the appropriateness of acquiring or building specific assets. Participants will illustrate the scope of their models by presenting scenarios where the models apply and by integrating the model into product line development patterns.

Models should address all or some of the following topics:

- Product lines introduce extra complexity in software development but offer high returns: What are the tradeoffs and when should software firms decide to go for a product line approach?
- What are the success and failure factors for introducing software product lines in organizations?
- Is it possible to predict when product line investment pays in a specific domain and environment?
- How can the benefits of a product line's success be quantified, taking into account nonfinancial factors?
- What are the effects of sustainment on the long-term benefits offered by the product line approach?

Participation

We would like to invite those working on improvements to state-of-the-art methods, tools, and technology in these areas to make presentations and to lead follow-up discussions. Please forward a short position paper to the organizers on one or more of the workshop topics listed below.

All the attendees of SPLC 2004 are welcome to participate in this workshop.

Workshop Format

The workshop will be a full-day workshop with at least six presentation sessions. Each presentation will outline work in a particular area as a starting point for discussion. Tentative presentation topics include

- What scenarios are necessary to represent the spectrum of business decisions to which the model can be applied?
- What are the strengths and shortcomings of existing models? How do we know if a model is correct and complete? Can it be enhanced?
- What are the quantifiable benefits that come from applying a product line approach? How are they modeled?
- How can we develop information to form a cost-benefit analysis? Will the model's representation be sufficiently intuitive for product line personnel to easily produce results and make those results understandable by managers and technicians alike?
- What are the attributes of a business case for adopting a product line approach to development? How can the model be sufficiently flexible to be useful in answering a wide range of questions that the business case must address?
- What tool support is in use or under development to support the model, and what types of tools are desired?

Submissions

To participate, please send your position paper to the organizer by June 4, 2004.

Quality Assurance in Reuse Contexts

Ronny Kolb, Fraunhofer Institute for Experimental Software Engineering (IESE) Sauerwiesen 6, D-67661 Kaiserslautern, Germany
John D. McGregor, Dept. of Computer Science, Clemson University Clemson, SC, USA
Dirk Muthig, Fraunhofer Institute for Experimental Software Engineering (IESE) Sauerwiesen 6, D-67661 Kaiserslautern, Germany

Workshop code: contact workshop organizer for workshop code.

The systematic, large-scale reuse of software development artifacts over multiple products is a promising approach to address today's software development problems and to make the development process more efficient. Recently, reuse-based software development paradigms such as component-based development and software product lines have increasingly received attention as they promise-and have shown-to shorten the development time of software systems and to reduce development and maintenance costs. To achieve the promised improvements, however, high-quality artifacts intended for reuse are required. Thus, more than for traditional software development, quality assurance becomes a crucial part of every reuse-based development effort. However, a number of specifics caused by software reuse (such as the variable usage of components or genericity of artifacts) must be faced during quality assurance. To enable an organization to fully experience the expected efficiency gain through reuse, therefore, a quality assurance approach is required that enables the validation of products built from reusable artifacts as effective and as efficient in a non-reuse context.

Despite the criticality of quality assurance and the special problems caused by reuse-based software development, research in the field of software product lines and component-based development has focused primarily on analysis, design, and implementation to date. Only very few results address the quality assurance problems and challenges that arise in a reuse context. Therefore, with the growing acceptance of reuse-based development paradigms such as software product lines, effective and efficient methods and techniques for ensuring the quality of reusable artifacts and products built by reusing existing artifacts are required.

The aim of the workshop is to establish a forum for the successful exchange of experience and ideas among practitioners and researchers to improve the state of the art and state of the practice in quality assurance for product lines and other reuse-based development approaches. The workshop will provide an opportunity to exchange views, experiences, and lessons learned, to advance ideas, and to discuss recent work and work in progress on topics dealing with quality assurance for software artifacts intended for reuse and products built using reusable artifacts. It intends to bring together researchers and practitioners from both academia and industry to share ideas on the foundations, techniques, methods, strategies, and tools of quality assurance for reuse-based software development paradigms.

Software Variability Management for Product Derivation - Towards Tool Support

Tomi Männistö, Helsinki University of Technology, Software Business and Engineering Institute, P.O. Box 9210, FI-02015 HUT, Finland
Jan Bosch, University of Groningen, Department of Computing Science, P.O. Box 800, NL-9700 AV Groningen, The Netherlands

Workshop code: contact workshop organizer for workshop code.

Software product lines aim at providing the means for achieving large software variability in an effective manner. However, systematic methods and tools are needed for describing and managing the variability so that large variability can be supported and an effective means for deriving product instances can be achieved.

Earlier workshops (at SPLC2 02, Groningen 03, ICSE03) on variability management have provided an initial understanding of the area and formed a basis for managing the variability of software product lines. Relevant results and lessons learned can also be found from traditional products (mechanical and electronic). In particular, the field of product configuration, which is an area using techniques of artificial intelligence, has recently started dealing with configuring software products that exhibit very large variability. The topic of software product configuration has been addressed in various workshops on configuration (e.g., in connection to the ECAI00, IJCAI01, ECAI02, IJCAI03 conferences).

Furthermore, initial computer-based tools or prototypes for managing variability in software product lines have already been demonstrated, and some projects addressing tool support exist in both industry and academia. Software variability management is thus moving towards tool support, although many theoretical and practical challenges remain to be resolved.

The workshop intends to bring together industrial developers and researchers who are building tools or working towards enabling tool support for software variability management. The theoretical underpinnings of variability modeling tools, such as modeling languages and requirements, are also within the interest of the workshop.

The workshop aims at moving from the current status of variability management towards tool support that would enable increasing the variability and customization possibilities of software product families in a feasible manner. In addition, the workshop promotes the transfer of knowledge between research and practice of traditional product families and software product families.

Solutions for Automotive Software Architectures: Open Standards, References, and Product Line Architectures

Stefan Ferber, Robert Bosch GmbH, Corporate Systems Engineering Process Group, Robert Bosch Str. 2, 71701 Schwieberdingen, Germany
Andreas Krüger, Audi AG, I/EE-93, 85049 Ingolstadt, Germany
Stefan Voget, Robert Bosch GmbH, Research and Development Department Software-Technology, Eschborner Landstrasse 130-132, 60489 Frankfurt, Germany
Matthias Weber, DaimlerChrysler AG, Research and Technology, 069/U119-RIC/SM, Alt-Moabit 96A, 10559 Berlin, Germany

Workshop code: WS0426

1. Goals

- Get an overview about existing and future software architectures for networked engine control units (ECUs) in automobiles.
- Give an overview about the international initiatives working on this topic.
- Get to know the key players.

2. Motivation

More and more product lines spread in the automotive industry. This is true for vehicle manufacturers as well as for suppliers. If both sides—customer and supplier—use the product line technology, the interfaces between both stakeholders become more important. The answer of the automotive industry to this challenge is the development of open standard architectures [3]. Several initiatives currently follow this path; for example, the ITEA/EAST-EEA [1] and AUTOSAR [2] Web sites.

During the workshop, we want to exchange experiences from the international activities and start an informal network. Such an information exchange should improve the product line technology for the automotive industry.

3. Organization of Workshop

All the attendees of SPLC 2004 are welcome to participate in this workshop. You are also welcome to inform yourself without a submitted paper.

The workshop splits into two parts.

1. a presentation that includes an overview of
    - activities on software architectures in Europe, Asia, and North America—at least one presentation

from each region
    - company-specific "standard" architectures—at least one presentation from an original equipment manufacturer (OEM) and one from a supplier
2. working groups. This part of the workshop incorporates a platform for group work for the participants' disposal. The number of groups will depend on the number of participants. At a minimum, the following topics will be prepared. Each one could be worked out by several groups:
    - commonalities and variants in the presented architectures. These groups should prepare a list that summarizes the given presentations on an harmonized and comparable level.
    - requirements on a global informal network for the exchange of information. These groups should collect reasons for such a network, organizational alternatives, alternatives for the kind of cooperation in such a network, and other items around this topic.

4. Expected Workshop Outputs

- a set of comparable "standard" architectures. This should increase the understandings between the architects and lead the discussion in the community to a harmonized level.
- list of points to be discussed and worked on in the informal network. Workshop participants should determine who should participate in such networks in the future. The experiences of the organizers in the last few years have shown that such a network is not unrealistic in the automotive community.

References

1. AUTOSAR: www.autosar.org
2. ITEA/EAST-EEA: www.east-eea.net
3. Schäuffele, J; Zurawka, T.: Automotive Software Engineering. Vieweg, 2003.

Workshop Attendance

If you are interested in attending a specific workshop, please follow the invitation criteria provided with the workshop and/or contact the workshop organizer(s).

The Third Software Product Line Conference

SPLC 2004

Panels

Charles Krueger
BigLever Software
Panels Chair

SPLC 2004 will host three conference panels:

Product Line Binding Times: What You Don't Know Can Hurt You
Avoiding, Surviving and Prevailing Over Pitfalls in Product Line Engineering
How Can Testing Keep Pace with Accelerated Development in Software Product Line Engineering?

Product Line Binding Times: What You Don't Know Can Hurt You
Panel Moderator: Charles W. Krueger, BigLever Software, Austin, TX, USA

Panelist
Dale Churchett, Salion, USA
Jan Bosch, University of Groningen, The Netherlands
Jim Snyder, Coremetrics, USA
Rob van Ommering, Philips, The Netherlands
Krzysztof Czarnecki, University of Waterloo, Canada
Joe Moore, Engenio Information Technologies (formerly LSI Logic Storage Systems), USA

Abstract
Choosing the right variation binding time–the point in the software engineering process at which decisions for variation instantiation are made–is often one of the most critical, most contentious, and least well understood issues for software product line organizations. This panel explores some of the fundamental binding time issues and insights based on the research, best practices, and real world experiences of the panelists.

Overview
One of the most critical, most contentious, and least well understood issues in a software product line approach can be the selection of the optimal binding time(s) and associated binding mechanism(s). There are often different agendas for the different stakeholders, plus preconceived and possibly inaccurate ideas about software product line engineering that can lead to strong differences of opinions about how to go about implementing a software product line. The goal of this panel is to expose the fundamental issues, myths, and hidden agendas surrounding binding time selection.

The essential distinction between software product line engineering and conventional software engineering is the

presence of variation in some of the software assets. In the early stages of the engineering process in a software product line approach, software assets contain *variation points* that represent unbound options about how the software will behave. At some point during the production process, the decision model is used to select among the options for the variation points. Then the behavior of the variation point in the final product is fully specified.

The time at which the decisions for a variation point are bound is referred to as the *binding time*, and the means by which the decisions are used to instantiate the variation points is referred to as the *binding mechanism*. Examples of binding times include source reuse, development, source instantiation, build, package, install, start-up, and execution time. Examples of different binding mechanisms include language *templates*, manual production, preprocessors, off-the-shelf product line tools, build scripts, CM *views*, installers, *config files*, and language conditionals.

Some of the questions and issues addressed by the panel include
1. What makes binding time selection a critical decision?
2. What are the criteria for selecting among different binding times? How do you identify the optimal choice?
3. When is it appropriate to use multiple binding times?
4. What are the different stakeholder agendas, and which ones can be detrimental?

Avoiding, Surviving, and Prevailing Over Pitfalls in Product Line Engineering (panel slides)
Panel Moderator: Charles W. Krueger, BigLever Software, Austin, TX, USA

Panelist
Jim Dager, Cummins, USA
Ulf Olsson, Ericsson, Celsius Tech (now Saab Tech)
Anders Heie, Nokia, USA
Martin Verlage, MARKET MAKER, Germany
Bill Hetrick , Engenio Information Technologies (formerly LSI Logic Storage Systems), USA

Abstract
In this panel, pioneers from some of the software product line field's best-known success stories share their experiences on how to detect potential pitfalls, how to avoid looming pitfalls, how to survive pitfalls that are unwittingly encountered, and how to turn the negative aspects of pitfalls into positive advantages.

Overview
It is inevitable that with every software product line success story, *a little rain must fall*. In retrospect, stories about the trials and tribulations that arise along the way to a software product line success can provide insight, guidance, education, portent, inspiration, and even entertainment.

This panel is made up of key leaders from well-known software product line success stories, sharing how they

- avoided the pitfalls that they were insightful enough–or lucky enough–to perceive before it was too late
- survived the pitfalls that surprised them along the way and how, in hindsight, they might have detected and avoided them
- prevailed over unavoidable pitfalls, turning negatives that could have doomed their efforts into positives that worked in their favor

The audience is encouraged to ask the panel for insight about specific pitfalls they have encountered.

How Can Testing Keep Pace with Accelerated Development in Software Product Line Engineering?
Panel Moderator: Birgit Geppert, Avaya Labs Research, Basking Ridge, NJ, USA

Panelists:
Carl Shaulis, Salion, USA
Henry Muccini, University of L'Aquila, Italy
Tim Trew, Philips Research Laboratories, UK
Claudia Fritsch, Bosch, Germany
Ronny Kolb, Fraunhofer Institute, Germany

Abstract
This panel explores the topic of software product line testing. Techniques for accelerated product development are widely published, but what are the implications to product testing in a software product line organization?

Overview
Although many organizations have reported ways in which software products can be created at a greatly accelerated pace using software product line techniques, little has been reported about the testing of those products. How do the test groups within these organizations keep pace with the accelerated product development?

This critical question is asked frequently by organizations considering the move to software product line engineering, particularly as they do their return-on-investment analysis. Without a convincing argument that testing can achieve similar accelerators, testing looms as the limiting bottleneck, making it appear that the test group must scale its expensive resources to support brute-force testing of all the new products rather than scaling the efficiency of its existing resources.

This panel brings together software product line test practitioners and researchers to shed light on this important topic and to address the following questions and more.

1. Is brute-force scaling of testing resources sufficient for software product line testing? Is it necessary?
2. The two guiding principles in software product line development are (a) capitalize on commonality and (b) formally manage variation among the products. Do the same principles guide software product line testing?
3. Conversely, are there other and possibly more important guiding principles that are unique to product line testing?
4. Is the variation space for development and testing the same? Can we leverage results from earlier development phases (such as decision modeling) for testing?
5. Are there additional problems that come up when dealing with variabilities that might even hinder/slow down testing?
6. *Domain engineering* of core assets is the prominent activity in software product line development, with significantly less effort dedicated to *application engineering* of the individual products. Intuitively, it seems that testing requires the opposite balance: more effort on integration and system testing than on unit testing. Is this true? How do you find the appropriate balance?
7. How much does improved product quality reduce the testing burden?

The Third Software Product Line Conference

SPLC 2004

Technical Demonstrations

Dale Churchett
Salion, Inc.
Demonstrations Chair

SPLC 2004 will host seven technical demonstrations:

Tools Supporting Domain-Specific Languages
Jack Greenfield - Microsoft Corporation, Redmond, WA, USA

Microsoft will present a demonstration of tools for Software Factories, which are highly automated software product lines, as described in the book titled *Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools* to be published soon by John Wiley and Sons. The demonstration will show three tools based on domain-specific languages (DSLs) and a suite of tools for developing them. DSLs play a key role in Software Factories by capturing meta-data about product family members and automating product-life-cycle processes. By making DSL design and implementation faster, cheaper, and easier, Microsoft is seeking to drive down the cost of product line development. That, in turn, will enable user organizations with domain knowledge to extend and customize canonical Software Factory templates to rapidly build Software Factories that automate the development of custom product families. For more information, see http://www.softwarefactories.com.

RequiLine: A Requirements Engineering Tool for Software Product Lines
Thomas Van Der Massen - Research Group Software Construction. University of Aaachen, Germany

Modeling variability is one of the most important tasks during the analysis phase of a software product line. Domain analysis and requirements elicitation will bring up a huge amount of requirements and dependencies between product characteristics. Feature modeling is a suitable approach as it provides a flexible formalism to express domain and product characteristics but is complex and requires tool support. RequiLine is a requirements

engineering tool prototype that supports the management of requirements and feature models equally.

MetaEdit+ metaCASE tool for Domain-Specific Modeling and Product Generation
Juha-Pekka Tolvanen - MetaCase, Jyvaskyla, Finland

Current modeling languages are based on the concepts of programming languages, leading to a poor mapping to product family characteristics and difficulties in leveraging the benefits and efficiencies of product family development. Domain-specific modeling languages fitting to the product family provide a viable solution. A modeling language (i.e., a meta-model) and variant generators are defined, based on the family characteristics.

The family-specific meta-model sets the variation space for possible models of variants and provides the basis for automated software production. The MetaEdit+ tool allows the definition and use of family-specific modeling languages and code generators. In the demonstration, the MetaEdit+ tool is described together with industrial examples from different domains such as mobile phones, home automation systems, and financial products.

Understanding Product Line Architectures Using Dependency Models
Neeraj Sangal & Ev Jordan - Lattix, Inc., Andover, MA, USA

Dependency models have been pioneered by systems engineering in order to model complex organizations, processes, and systems. Systems engineers have built Design Structure Matrices (DSMs) to represent complex systems and to reorganize systems for greater efficiency and modularity. A DSM is particularly useful for modeling software product line architectures because it is built around a precise decomposition of a system into its components. The simplicity and compactness of DSMs also makes it easy for various stakeholders to visualize the overall architecture. Lattix will demonstrate a new technique based on dependency models for understanding and managing large complex software systems.

ConIPF: Configuration in Industrial Product Families
John MacGregor - Robert Bosch GmbH, Frankfurt, Germany

ConIPF is a publicly funded project that is defining a methodology for the derivation of products (software systems) from industrial product lines. Ultimately, the methodology will be described in the book tentatively titled *Configuration in Industrial Product Lines: Challenges, Solutions, and Examples*. The methodology describes how to model features, product architectures, and hardware and software components so that structure-based configuration can be used to derive the products directly using available components and calibrate those components. The methodology also describes how to handle evolution.

Software Mass Customization with BigLever Software's GEARS
Charles Krueger - BigLever Software, Austin, TX, USA

This demonstration will illustrate the capabilities of BigLever Software's GEARS, a commercially available technology that supports software product line engineering through "software mass customization." The demo will benefit practitioners and managers who are responsible for establishing and maintaining software product lines and researchers who are interested in learning more about the practical issues of engineering software product lines.

Building the Buzz: www.softwareproductlines.com
Charles Krueger - BigLever Software, Austin, TX, USA

This demonstration will provide an overview of the community Web site for software product line practitioners, www.softwareproductlines.com. The site is devoted to the community of software engineers interested in using software product line approaches for developing their software. The objective of the demonstration is to make people aware of this resource, to solicit feedback, to encourage participation in developing the sites content, and to grow the sense of community "buzz" around the software product line field.

The Third Software Product Line Conference

SPLC 2004

Keynote Address

Jack Greenfield
Microsoft

Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools

Increasingly complex and rapidly changing requirements and technologies are pushing the limits of the current approach to application development. Because of these trends, stakeholders are increasingly turning to software product line practices to reduce risk, cost, and time to market while improving product quality through systematic reuse. Adopting organizations are starting to demand the kind of tool support available in Rapid Application Development (RAD) environments for software product lines. This talk describes a methodology developed at Microsoft called Software Factories. The goal of the methodology is to enable automation of life-cycle tasks in software product line contexts by integrating innovations in model-driven development (MDD), component-based development (CBD), and agile development methods. Software Factories is based on a four-part pattern for building patterns, models, frameworks, and tools for specific domains, such as user interface construction or database design. The key to the methodology is reducing the cost of implementing the pattern, making it cost-effective for narrower and more specialized domains, such as B2C application development and business process automation. The central concept is the software schema-a network of viewpoints describing the artifacts that constitute the members of a family of software products and identifying the patterns, languages, frameworks, tools, processes, and other assets used to build those artifacts. Mappings between viewpoints support artifact transformation and provide constraints on the development process that enable a scalable approach to agile development. By automating many aspects of the product development process, the Software Factories methodology provides a basis for industrializing software development and promotes the formation of software supply chains, paving the way for software mass customization.

Biography

Jack Greenfield is an architect for enterprise frameworks and tools at Microsoft. He was previously the chief architect of the Practitioner Desktop Group at Rational Software Corporation, and the founder and CTO of InLine Software Corporation. At NeXT, he was a key contributor to the Enterprise Objects Framework, now known as Web Objects from Apple Computer. A well-known speaker and writer, Mr. Greenfield is also coauthor of the book *Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools* published by John Wiley and Sons. He has contributed to the Unified Modeling Language (UML), the Java 2 Platform, Enterprise Edition (J2EE), and related Object Management Group (OMG) and Java Community Process (JCP) specifications. He holds a BS degree in Physics from George Mason University.

The Third Software Product Line Conference

SPLC 2004

Product Line Hall of Fame

Four new software product lines were inducted into the Hall of Fame at SPLC 2004: General Motors' Powertrain product line, Salion, Inc.'s product line of revenue acquisition management systems, Ericsson's AXE family of telecommunication switches, and Phillips' product line of software for high-end television sets.

A hall of fame serves as a way to recognize distinguished members of a community in a field of endeavor. Those elected to membership in a hall of fame represent the highest achievement in their field, serving as models of what can be achieved and how. Each Software Product Line Conference culminates with a session in which members of the audience nominate systems for induction into the Software Product Line Hall of Fame. These nominations feed discussions about what constitutes excellence and success in product lines. The goal is to improve software product line practice by identifying the best examples in the field. At the end of the deliberations, those systems elected by the majority of those present attain membership in the Hall of Fame.

The following product lines were inducted into the Software Product Line Hall of Fame during the First and Second Product Line Conferences.

The First Software Product Line Conference (SPLC1)

A-7E Operational Flight Program, U.S. Naval Research Laboratory

The A-7E operational flight program (OFP) is the software that assists the pilot of the Navy's A-7E aircraft to operate the airplane. The OFP was redesigned by the Software Cost Reduction project at the Naval Research Laboratory to show how to apply family-based software development principles in the development of a hard-real-time system. Commonalities and variabilities were explicitly identified starting in the requirements specification for the family and were a strong driving factor in the modular design of the OFP. The OFP design, including a modular structure, a process structure, and a uses relation, was explicitly created and documented to be an engineering model that others could follow. It has had a strong influence on the fields of both software engineering and product line engineering.

- Len Bass, Paul Clements, and Rick Kazman, *Software Product Lines: Practices and Patterns*, Addison Wesley, 1998, Chapter 3.

ShipSystem 2000, CelsiusTech Systems AB

ShipSystem 2000 is a family of naval shipboard comment and control systems produced by CelsiusTech Systems AB of Sweden since the late 1980s. Begun in 1985 as a business and technical response to two large contracts awarded simultaneously, ShipSystem 2000 is based on a robust architecture that was designed to handle both of those initial systems, as well as the more than 50 variants that followed. Family members include systems for ships from coastal corvettes to cruisers to submarines, for navies all over the world. These systems comprise 1-1.5 million source lines of code (SLOC) of Ada code, are hard-real-time, embedded, and safety critical. CelsiusTech has been able to slash production time, build more systems with fewer people, and increase quality. The story of ShipSystem 2000 was one of the first and most important case studies in successful software product line engineering.

- Len Bass, Paul Clements, and Rick Kazman, *Software Product Lines: Practices and Patterns*, Addison Wesley, 1998, Chapter 16.

Mobile Phones, Nokia

Nokia Mobile Phones produces a wide range of mobile phones. Currently 32 different phones are manufactured covering six different protocol standards, a wide variety of functional features and capabilities, different user interface designs, and many platforms and environments. The initial software architecture for this product line addressed variations in hardware, communication standards, and user interfaces; the product line was selected as "The Product of the Year" by *Business Week* and *Connect* magazines. The current architecture is component based in the client-server style. It allows separate service providers to be plugged in or taken out without restarting the system. This architecture supports both local and remote message passing, component management, task scheduling, and event control. Nokia Mobile Phones is the world's largest mobile phone manufacturer, and the company's leaders believe that software product line engineering has helped the company to reach that position.

- Slide Set: Global Software Product Lines and Infinite Diversity – Anders Heie

Owen Firmware Cooperative, Hewlett-Packard

Owen is a community of firmware development teams from Hewlett-Packard (HP) product divisions in two states in the USA; they produce firmware for a number of printers and printer/copier/scanner/fax devices. Participating teams contribute to the cooperative by producing assets conformant to the Owen architecture and benefit from other teams' contributions. Owen is unique because of its strong cultural aspects. A steering team, a firmware architect, a firmware asset lead, and "cooperative steward" roles provide the overall direction. There are cooperative operating principles, and members (while first and foremost turning out their own products) have explicit responsibilities to the co-op. Owen products have been produced using 1/4 of the staff, in 1/3 of the time, and with 1/25 the number of bugs of earlier products.

- Peter Toft, Derek Coleman, and Joni Ohta, "A Cooperative Model for Cross-Divisional Product Development for a Software Product Line" Patrick Donohoe (ed.) *Proceedings SPLC1*, Kluwer Academic Publishers, 2000.

The Second Software Product Line Conference (SPLC2)

Diesel Engine Software Product Line, Cummins, Inc.

Cummins, Inc., is the world's largest manufacturer of large diesel engines. Modern engines can contain over 100KSLOC of software to micro-control ignition to produce an optimum mix of power, economy, and emissions. In 1993, faced with the need to produce almost 20 new systems but with staff and resources available only for 6, Cummins changed the way it developed software and embraced the product line approach. Cummins' product line is a story of the extensive use of legacy software, strong processes, and a culture of intra-organizational cooperation.

Today the Cummins software product line covers 9 basic engine types ranging over 4-18 cylinders and 4-164 liters of displacement, with 12 kinds of electronic control modules, 5 kinds of processors, and 10 kinds of fuel systems. To date, 20 basic software builds have been parlayed into well over 1,000 separate products. Cycle time has been reduced from around 250 person-months to a few person-months. Quality and customer satisfaction are both up, and 15 of 15 projects are on track. Cummins estimates a productivity improvement of 3.6 and a return on investment (ROI) of 10:1 from the product line approach. The approach has also enabled Cummins to quickly enter and become successful in a related market area—namely, industrial diesel engines that power a variety of applications from rock crushers to ski lifts.

- P. Clements and L. Northrop, *Software Product Lines: Practices and Patterns*, Addison Wesley, 2001.

Telecommunication Switching System, Philips

The PKI Telecommunications Switching System (TSS) is a product family (product line) originating from the middle of the 1980s. PKI was a small player in the telecommunications world and had to survive by addressing a niche market. In particular, the TSS family had to serve a large variety of clients and regulations. The approach emphasized a component-based architecture; components were called "Building Blocks." The architecture consisted of a component-based framework in which plug-ins are available to tailor the system to the actual requirements. Moreover, aspects were defined for meeting quality requirements. For many aspects, automatic code generation was available. For other aspects, code guidelines were available, easing the burden of implementation.

The architecture of the system ensured that the system could be built and tested incrementally. The family was very successful in having a fast time to market and high reuse.

In 1994, Lucent bought PKI and did not continue the TSS family. The knowledge about the TSS system stayed within Philips, however, and the majority of the present-day product-family developments within Philips are still influenced by the TSS experiences.

- Frank van der Linden and Jürgen K. Müller: "Creating Architectures with Building Blocks," *IEEE Software*, Nov. 1995.
- Frank van der Linden, Jürgen K. Müller: Composing Product Families from Reusable Components, Bonnie Melhart, Jerzy Rozenblit (eds.), *Proceedings 1995 International Symposium and Workshop on Systems Engineering of Computer Based Systems*, IEEE, pp. 35 - 40 (1995).

- Jürgen K. Müller: “Integrating Architectural Design into the Development Process,” Bonnie Melhart and Jerzy Rozenblit (eds.), *Proceedings 1995 International Symposium and Workshop on Systems Engineering of Computer Based Systems*, IEEE, pp. 114 - 121 (1995).

- Jürgen K. Müller: “Feature-Oriented Software Structuring,” Proceedings *CompSAC'97*, pp. 552-555, (1997).
- Jan Gerben Wijnstra: Critical Factors for a Successful Platform-Based Product Family Approach, Gary J. Chastek (ed.) *Proceedings SPLC2*, Springer LNCS 2379, (2002).

5ESS Telecommunications Switch, Bell Labs / AT&T / Lucent

The 5ESS[tm] product line is a family of telephone switches that has an unparalleled reputation for reliability, quality, and performance. The switch was originally developed by AT&T Bell Labs and first put into commercial use in 1982. It is currently made by Lucent Technologies. The majority of local telephone switches in the U.S. today are still 5ESS switches. If you live in the U.S., most likely when you pick up the handset on your telephone, you are connected to a 5ESS switch.

Any particular switch in the product line is operated by approximately 10M SLOC. The software architecture reflected in that code has remained relatively stable at the subsystem level over a period of 20 years and was designed to accommodate a set of variabilities that can still be discerned by examining the architecture. In the early 1990s, some of the first applications of domain engineering to a large, complex system were accomplished successfully in the 5ESS software and documented in the software engineering literature. Domains such as switch maintenance, signaling, and traffic management showed productivity improvements of factors of 3 to 5 as a result.

- W. Howard, editor, “The 5ESS Switching System,” vol. 64, *AT&T Technical Journal*, July-August, 1985, Special Issue on the 5ESS Switch.

Bold Stroke Avionics Software Family, Boeing

The Bold Stroke Software Product Line comprises a wide range of artifacts required to create operational flight programs (OPFs) for a variety of Boeing military fighters, including a highly configurable architecture, application components, middleware framework, and development processes and tools. OPFs are mission-critical, distributed, real-time, embedded applications supporting the avionics and cockpit functions for the pilot. A well-defined software architecture and carefully designed approaches to handle commonality and variability were crucial to the success of this product line. The architecture is heavily based on and expressed via object-oriented patterns. These patterns were leveraged to convey both the architecture and its rationale to a large community of software engineers previously experienced primarily with military standard assembly language systems. The product line exploits commercial standards, technologies, and products as much as possible, using a commercial real-time operating system and an open source real-time object request broker (ORB)—the ACE ORB—developed in partnership with Washington University in St. Louis. The Bold Stroke Software Product Line is the foundation for an increasing number of production and research programs including

several funded by the Air Force Research Laboratory and the Defense Advanced Research Projects Agency.

- Sharp, David C., "Reducing Avionics Software Cost Through Component Based Product Line Development," Patrick Donohoe (ed.) *Proceedings SPLC1*, Kluwer Academic Publishers, 2000.

- Doerr, Bryan S., and Sharp, David C., "Freeing Product Line Architectures from Execution Dependencies," Patrick Donohoe (ed.) *Proceedings SPLC1*, Kluwer Academic Publishers, 2000.
- Popp, Timothy J., "Software Architecture Development for Product Line Software," *AIAA/ IEEE Digital Avionics Systems Conference*, October 1999.

The MERGER Software Product Line (MARKET MAKER Software AG)

MARKET MAKER Software AG, Kaiserslautern, Germany, provides Europe's most popular stock market software. Since 1989, its products have allowed the stock market to be tracked and analyzed. In 1999, MARKET MAKER decided to launch an internet-based version of its product, using the functionality of its desktop products as the engine to power other companies' financial Web sites. This kind of system has to integrate with the customers' databases and other content-producing software, run on "who knows what" kind of computing platforms and servers, satisfy human-user performance requirements, and be tailored to show exactly the kind of data, in exactly the kind of charts, in exactly the kind of form required by each particular customer's Web site. That is, the product must be flexible, widely tailorable, deliverable in a very short amount of time, and producible by a very small development staff.

For these reasons, MARKET MAKER decided to plan the Internet versions right from the beginning as a software product line called MERGER. The result is a 520K SLOC system that meets all those requirements and more. Six people (two of whom were part-time) worked for about a year to produce the core system, from which instantiated products are turned out. Each product in the family must be built to the client's specifications and installed and tested on the client's own platform. Because of its systematic product line approach, MARKET MAKER can set up such systems in a few days. In the early days of the product line, this short time to market was the major advantage of MARKET MAKER over its competitors. In the current bad economic times, MARKET MAKER can survive because it has a small, efficient team that maintains its running systems.

- P. Clements and L. Northrop, *Software Product Lines: Practices and Patterns*, Addison Wesley, 2001.

- C. Gacek, P. Knauber, K. Schmid, and P. Clements. Successful Software Product Line Development in a Small Organization. A Case Study, Technical Report, Fraunhofer Institut for Experimental Software Engineering (IESE), 013.01/E, 2001.

Criteria for Election to the Software Product Line Hall of Fame

Members of the software product line hall of fame should serve as models of what a software product line should be,

exhibiting most or all of the following characteristics:

- The family that constitutes the product line is clearly identified, that is, there is a way to tell whether a software system is a member of the product line, by applying either a known rule or a known enumeration.
- The family that constitutes the product line is explicitly defined and designed as a product line, that is, the commonalities and variabilities that characterize the members of the product line are known, and there is an underlying design for the product line that takes advantage of them.
- The product line has had a strong influence on others who desire to build and evolve product lines, and has gained recognition as a model of what a product line should be and how it should be built. Others have borrowed, copied, and stolen from it when creating their product lines or expounding ideas and practices for creating product lines.

- The product line has been commercially successful.
- There is sufficient documentation about the product line that one can understand its definition, design, and implementation without resorting solely to hearsay.

The Third Software Product Line Conference

SPLC 2004

Program

SPLC 2004 pays special attention to three critical issues in product line engineering: (1) architecture, (2) quality assurance, and (3) business and economics. For each of these topics, we've created special tracks that include dedicated tutorials, workshops, panels, and paper sessions.

The following links provide multiple views of the SPLC 2004 program.

1. Conference Program
2. Conference Program showing all tracks
3. Track information (by day)
4. Architecture for Product Lines track only
5. Business and Economic Issues track only
6. Quality Assurance track only

The Third Software Product Line Conference

SPLC 2004

Program Committee

Mark Ardis, Rose Hulman Institute of Technology, USA
Wolfram Bartussek, ProSys, Germany
Joe Baumann, Hewlett-Packard, USA
Günter Böckle, Siemens Research, Germany
Jan Bosch, University of Groningen, The Netherlands
Sholom Cohen, Software Engineering Institute, USA
Krzysztof Czarnecki, University of Waterloo, Canada
Premkumar Devanbu, University of California at Davis, USA
Ed Dunn, Naval Undersea Warfare Center, USA
Stuart Faulk, University of Oregon, USA
Stefan Ferber, Robert Bosch, Germany
Cristina Gacek, University of Newcastle upon Tyne, Great Britain
Joe Gahimer, Cummins, USA
Birgit Geppert, Avaya Laboratories, USA
Jack Greenfield, Microsoft Corp., USA
Anders Heie, Nokia, USA
Mei Hong, Peking University, China
Peter Knauber, Mannheim University of Applied Sciences, Germany
Robyn Lutz, Jet Propulsion Laboratory, USA
Dirk Muthig, Fraunhofer IESE, Germany
Dale Peterson, Convergys, USA
Klaus Pohl, University of Essen, Germany
Scott Preece, Motorola, USA
Andre van der Hoek, University of California at Irvine, USA
Frank van der Linden, Philips Medical Systems, The Netherlands
Jay van Zyl, SystemicLogic, South Africa
Martin Verlage, Market Maker Software, Germany
Rich Zebrowski, Motorola, USA

Software Product Lines
Third International Conference, SPLC 2004, Boston, MA, USA, August 30-September 2, 2004, Proceedings
Series: Lecture Notes in Computer Science , Vol. 3154
Nord, Robert L. (Ed.)
2004, XIV, 335 p., Softcover
ISBN: 978-3-540-22918-6
Online version available

Usually dispatched between 3 to 5 business days

$74.95

About this book | Table of contents | Sample pages

About this book

This book constitutes the refereed proceedings of the Third International Software Product Line Conference, SPLC 2004, held in Boston, MA, USA in August/September 2004.

The 18 revised full technical papers presented together with a keynote abstract and summaries of panels, tutorials, and workshops were carefully reviewed and selected for inclusion in the book. Organized in sections on business, architecture, and quality assurance, the papers address topics ranging from how to start a software product line in a company, to case studies of mature product lines and the technology used, to test strategies of product lines, to strategies and notations for creating product line architectures, and to the importance of binding times in creating product lines.

Written for:
Reasearchers and professionals

Keywords:
- component-based systems
- concerns
- product line architectures
- product line tools
- reengineering
- software development
- software engineering
- software management
- software modeling
- software process
- software product lines

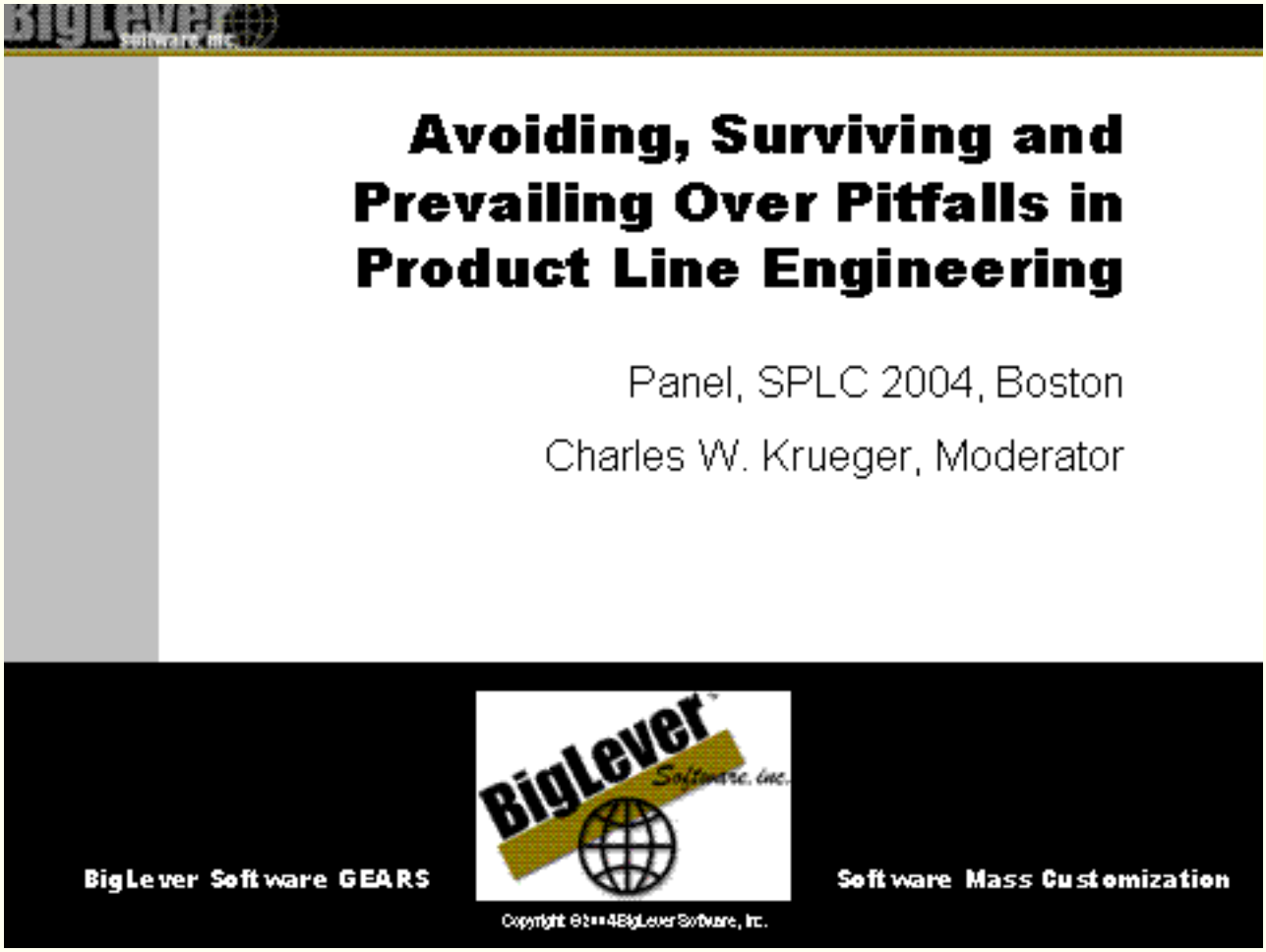
Print
Recommend to others
Download Flyer

E-content
Online Version Click here !

All books by this editor
Nord, Robert L.

Related subjects
General Computer Science
Programming, SWE & Operating Systems

- software projects
- uml



Next Exit Slide Show

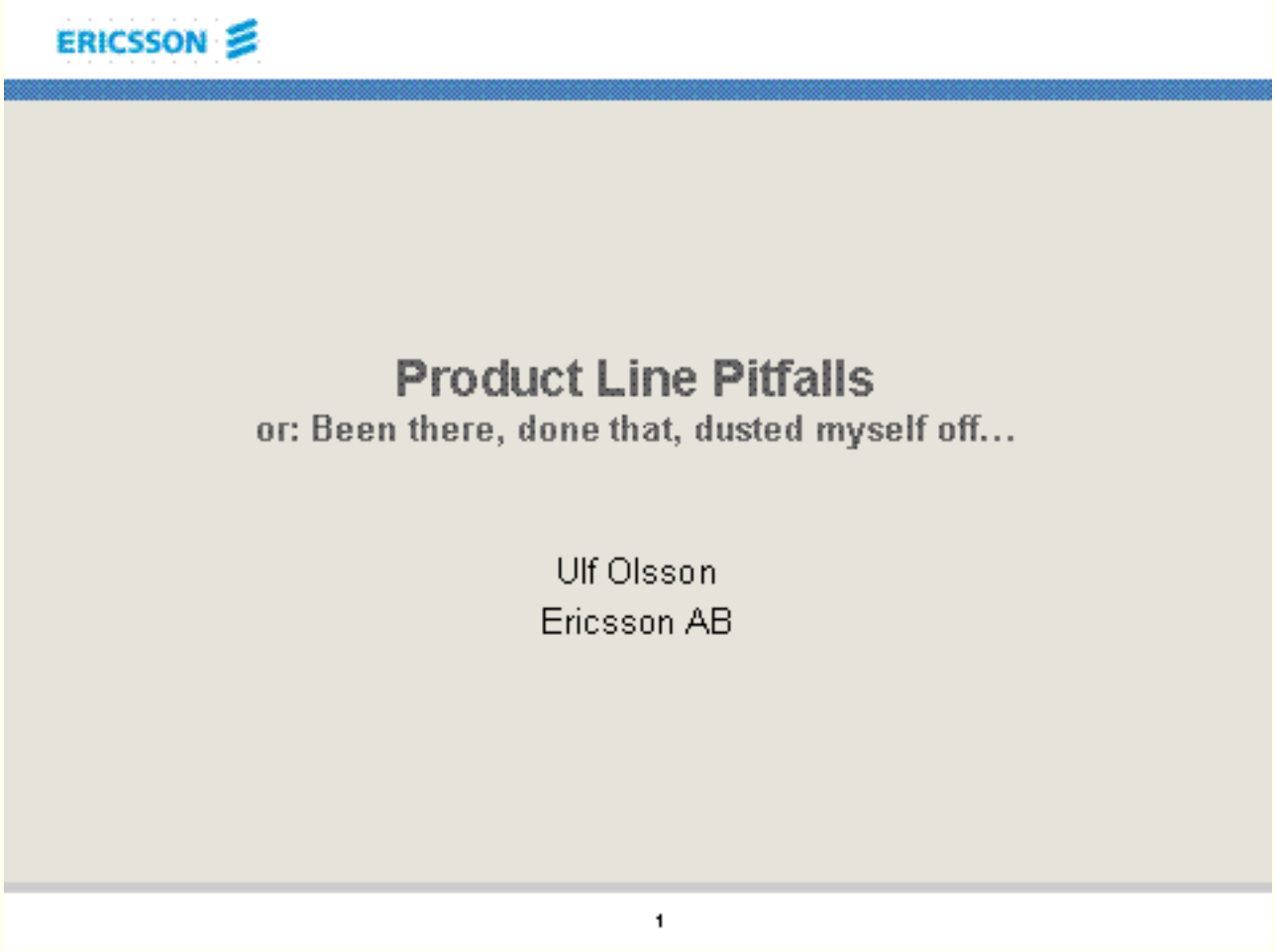
1 2 3 4 5 6 7 8

Cummins, Inc.
Avoiding, Surviving and Prevailing Over Pitfalls in Product Line Engineering

Jim Dager
Director, CORE Controls
Cummins, Inc.
jim.c.dager@cummins.com

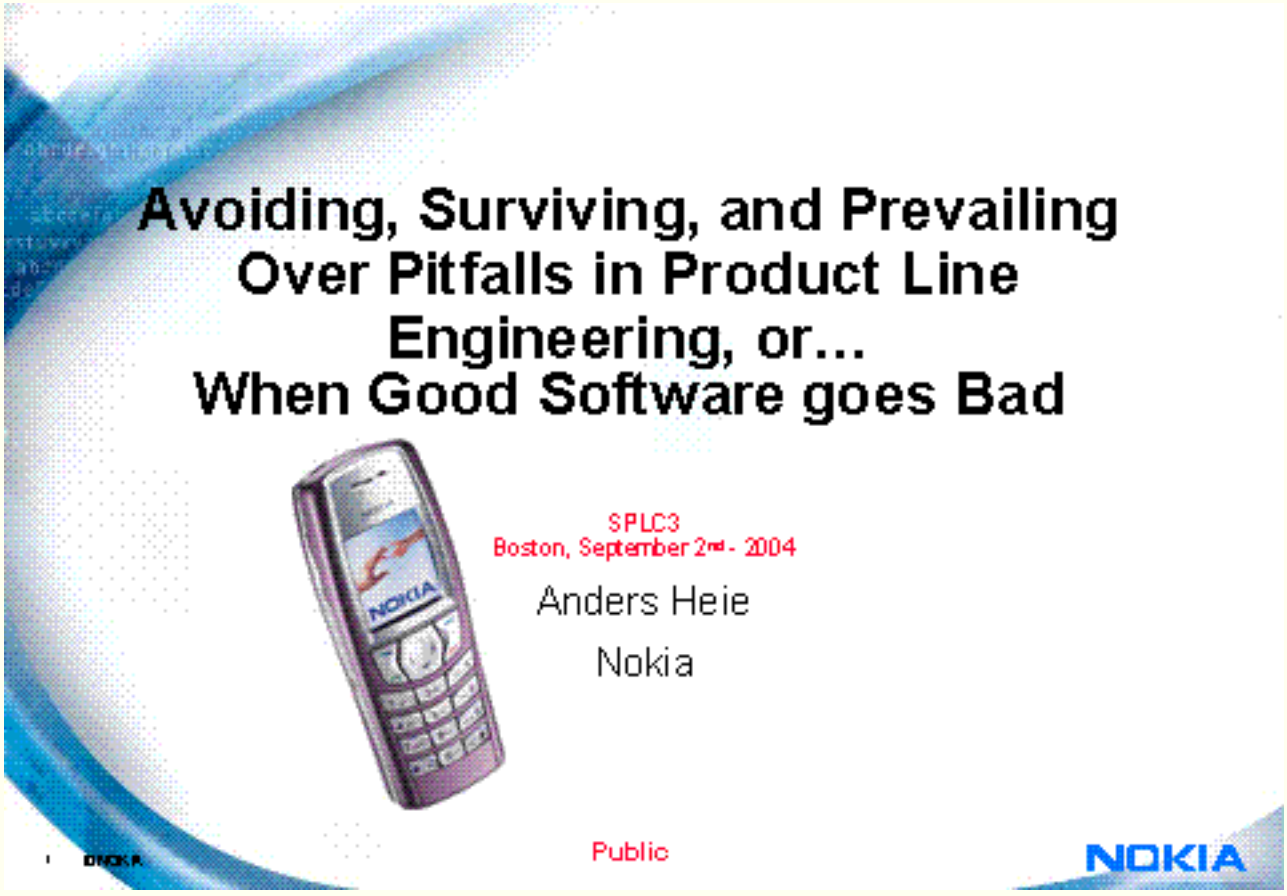
Next Exit Slide Show

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30



Next Exit Slide Show

1 2 3



Next Exit Slide Show

1 2 3 4 5 6

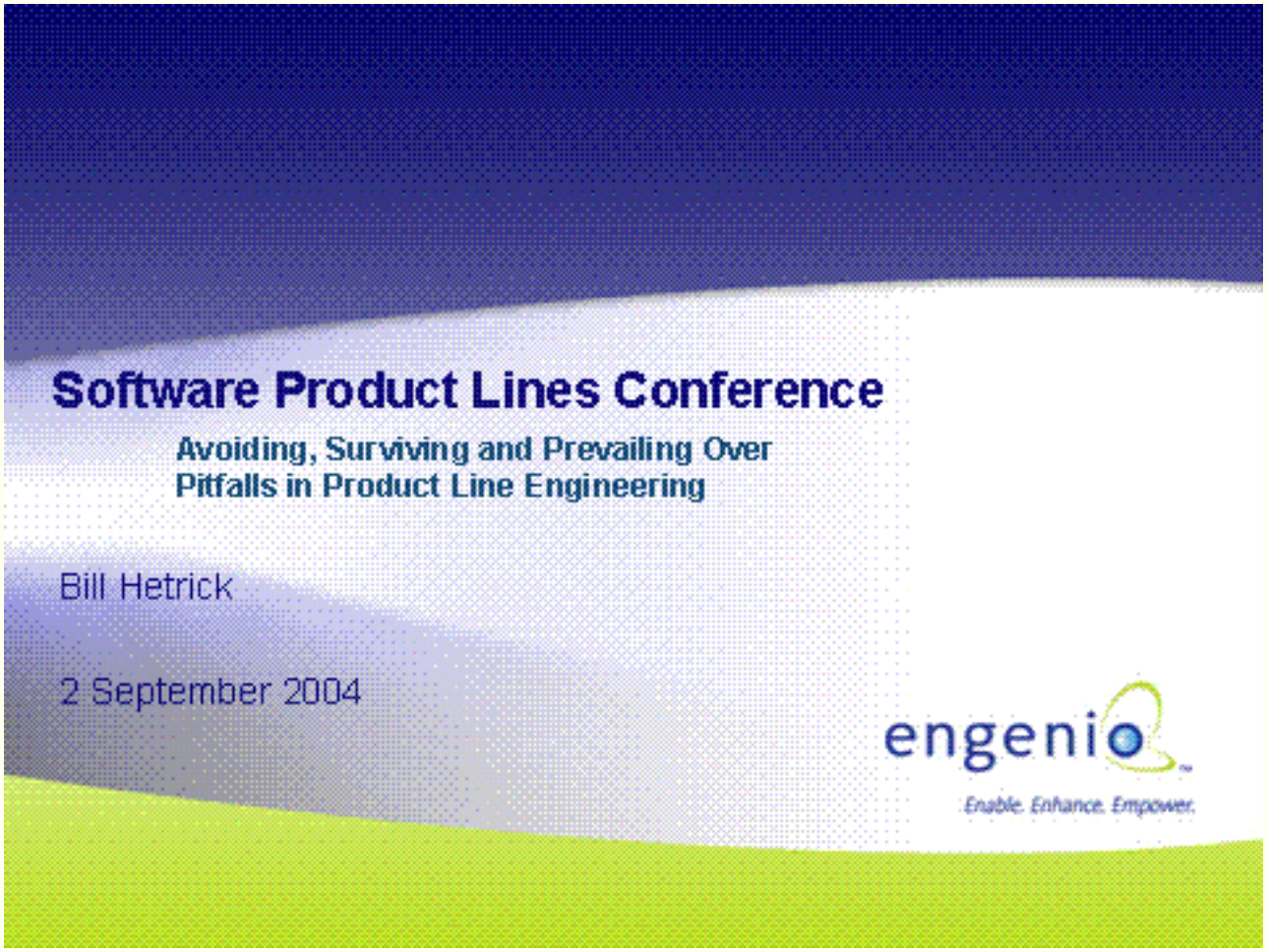
Avoiding, Surviving, and Prevailing Over Pitfalls in Product Line Engineering

Panel Session
3rd SPLC
Boston, MA
September 2, 2004

Martin Verlage
Director Data & Services
MARKET MAKER Software AG
Kaiserslautern, Germany
m.verlage@market-maker.de

Next Exit Slide Show

1 2 3 4 5 6 7



Next Exit Slide Show

1 2 3 4 5 6 7 8

The Third Software Product Line Conference

Conference Program

SPLC 2004

Day 1 - Monday, August 30th, 2004

7:30 – 8:30	Continental Breakfast				
8:30 – 12:00	**Workshop:** Solutions for Automotive Software Architectures: Open Standards, References, and Product Line Architectures. S. Ferber, A. Krüger, S. Voget, M. Weber	**Workshop:** Quality Assurance in Reuse Contexts R. Kolb, J. D. McGregor, and D. Muthig	**Tutorial:** An Introduction to Software Product Lines L. Northrop and P. Clements	**Workshop:** SPLY'E: The First Software Product Lines Young Researchers Workshop B. Geppert, I. John, G. Lami	**Tutorial:** Architecture-Centric Software Engineering J. Bosch	**Tutorial:** Starting Product Lines (I) — Systematic Product Line Planning and Adoption K. Schmid and I. John
10:00 – 10:30	Refreshment Break					
12:00 – 1:30	Lunch					
1:30 – 5:00	**Workshop Continued**	**Workshop Continued**	**Tutorial Continued**	**Tutorial:** Developing a Measurement Program for Software Product Lines S. Cohen, D. Zubrow, G. Chastek	**Tutorial:** Software Variability Management J. Bosch	**Tutorial:** Starting Product Lines (II) — Product Line Analysis and Modeling I. John and K. Schmid
3:00 – 3:30	Refreshment Break					

Day 2 - Tuesday, August 31st, 2004

7:30 – 8:30	Continental Breakfast						
8:30 – 12:00	**Workshop:** SPLiT - Workshop on Software Product Line Testing B. Geppert, C. Krueger, J. Li	**Tutorial:** Product Line Analysis G. Chastek and P. Donohoe	**Tutorial:** Adopting Software Product Lines L. Northrop and L. Jones	**Workshop:** Software Variability Management for Product Derivation - Towards Tool Support T. Männistö and J. Bosch	**Workshop:** Modeling Business Issues of Software Product Lines S. Cohen	**Tutorial:** Product Line Architectures for Global Software Development D. J. Paulish, R. Pichler, and W. Kuhn	**Tutorial:** Quality Assurance for Software Product Lines R. Kolb and D. Muthig
10:00 – 10:30	Refreshment Break						
12:00 – 1:30	Lunch						
1:30 – 5:00	**Workshop Continued**	**Tutorial:** Industrial Strength Software Product line Engineering J. Klein, D. Hill, and D. Weiss	**Tutorial:** Generative Software Development K. Czarnecki	**Workshop Continued**	**Workshop Continued**	**Tutorial:** Using Domain-Specific Languages, Patterns, Frameworks, and Tools to Assemble Applications J. Greenfield	**Tutorial:** Designing Software Product Lines with the Unified Modeling Language (UML) H. Gomaa
3:00 – 3:30	Refreshment Break						
5:30 – 7:30	**Social Reception**						

Day 3 - Wednesday, September 1st, 2004

8:00 – 8:30	Continental Breakfast		
8:30 – 10:00	**Session 1**		
	8:30 – 9:00	Welcome	
	9:00 – 10:00	**Keynote** *Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools* Jack Greenfield, Microsoft	
10:00 – 10:30	Refreshment Break		
10:30 – 12:00	**Session 2** **Business: Experiences in Introducing Product Line Engineering** Session Moderator: Martin Verlage, Market Maker Software AG		
	10:30 – 11:00	*Governing Software Product Lines and Reorganizations* Truman M. Jolley, Boeing Commercial Airplanes David J. Kasik, Boeing Commercial Airplanes Tammy R. Ben, Boeing Commercial Airplanes	
	11:00 – 11:30	*Software Product Line Support in Coremetrics OA2004* James Snyder, Coremetrics, Inc. Harry Lai, Coremetrics, Inc. Shirish Reddy, Coremetrics, Inc. Jimmy Wan, Coremetrics, Inc.	
	11:30 –12:00	*Introducing the Product Line Approach at Bosch Gasoline Systems: Experiences and Practices* Mirjam Staiger, Robert Bosch GmbH Christian Tischer, Robert Bosch GmbH Birgit Boss, Robert Bosch GmbH Andreas Müller, Robert Bosch GmbH Oliver Pertler, Robert Bosch GmbH Wolfgang Stolz, Robert Bosch GmbH Stefan Ferber, Robert Bosch GmbH	
12:00 – 1:00	Lunch		
1:00 – 2:30	**Session 3**		
	Workshop Report Session Moderator: Klaus Schmid, Fraunhofer IESE		**Architecture: Panel** Session Moderator: Charles Krueger, BigLever Software
	1:00 – 2:30	Workshop organizers report on results	1:00 – 2:30 *Product Line Binding Times: What You Don't Know Can Hurt You*
2:30 – 3:00	Refreshment Break		
3:00 – 4:30	**Session 4** **Architecture: Product Creation, Part I** Session Moderator: Stefan Ferber, Robert Bosch		
	3:00 – 3:30	*Four Mechanisms for Adaptable Systems: A Meta-Level Approach to Building a Software Product Line* Claudia Fritsch, Robert Bosch GmbH Burkhardt Renz, University of Applied Sciences Gießen-Friedberg	
	3:30 – 4:00	*Automatic Generation of Program Families by Model Restrictions* Andrzej Wasowski, IT University of Copenhagen	
	4:00 – 4:30	*Software Product Lines in ArchJava* Sebastian Pavel, Ecole des Mines de Nantes - INRIA Jacques Noyé, Ecole des Mines de Nantes - INRIA Jean-Claude Royer, Ecole des Mines de Nantes - INRIA	
5:30 – 8:30	**Demonstrations**		
	5:30 & 7:00 *RequLine: A Requirements Engineering Tool for Software Product Lines* Thomas Van Der Massen 6:00 & 7:30 *MetaEdit+ metaCASE tool for Domain-Specific Modeling and Product Generation* Juha-Pekka Tolvanen	5:30 & 7:00 *ConIPF: Configuration in Industrial Product Families* John MacGregor 6:00 & 7:30 *Understanding Product Line Architectures Using Dependency Models* Neeraj Sangal & Ev Jordan	5:30 & 7:30 *Software Mass Customization with BigLever Software's GEARS* Charles Krueger 6:00 & 7:00 *Tools Supporting Domain-Specific Languages* Jack Greenfield 6:30 & 8:00 *Building the Buzz: www.softwareproductlines.com* Charles Krueger
7:00 – 9:00	**Birds-of-a-Feather Sessions**		

Day 4 - Thursday, September 2nd, 2004

8:00 – 8:30	Continental Breakfast			
8:30 – 10:00	**Session 1**			
	Business: Frameworks for Introducing Product Line Engineering Session Moderator: Dirk Muthig, Fraunhofer IESE		**Quality Assurance: Panel** Session Moderator: Charles Krueger, BigLever Software	
	8:30 – 9:00	*Software Product Family Evaluation* Frank van der Linden, Philips Medical Systems Jan Bosch, University of Groningen Erik Kamsties, University of Duisburg-Essen Kari Känsälä, Nokia Research Center Henk Obbink, Philips Research Laboratories	8:30 – 10:00	*How Can Testing Keep Pace with Accelerated Development in Software Product Lines?*
	9:00 – 9:30	*Practical Evaluation of Software Product Family Architectures* Eila Niemelä, VTT Technical Research Centre of Finland Mari Matinlassi, VTT Technical Research Centre of Finland Anne Taulavuori, VTT Technical Research Centre of Finland		
	9:30 – 10:00	*On the Development of Software Product Family Components* Jan Bosch, University of Groningen		
10:00 – 10:30	Refreshment Break			
10:30 – 12:00	**Session 2**			
	Architecture: Product Creation, Part II Session Moderator: Scott Preece, Motorola		**Quality Assurance: Evaluation and Planning** Session Moderator: Frank van der Linden, Philips	
	10:30 – 11:00	*Experiences in Software Product Families: Problems and Issues During Product Derivation* Sybren Deelstra, University of Groningen Marco Sinnema, University of Groningen Jan Bosch, University of Groningen	10:30 – 11:00	*Observations from the Recovery of a Software Product Family* Patricia Lago, Vrije Universiteit Amsterdam Hans van Vliet, Vrije Universiteit Amsterdam
	11:00 – 11:30	*A Feature-Based Approach to Product Line Production Planning* Jaejoon Lee, Pohang University of Science and Technology Kyo C. Kang, Pohang University of Science and Technology Sajoong Kim, Korea Software Institute	11:00 – 11:30	*Product Line Potential Analysis* Claudia Fritsch, Robert Bosch GmbH Ralf Hahn, Robert Bosch GmbH
	11:30 – 12:00	*COVAMOF: A Framework for Modeling Variability in Software Product Families* Marco Sinnema, University of Groningen Sybren Deelstra, University of Groningen Jos Nijhuis, University of Groningen Jan Bosch, University of Groningen	11:30 – 12:00	*Generalized Release Planning for Product Line Architectures* Louis J. M. Taborda, Macquarie University

12:00 – 1:00	Lunch			
1:00 – 2:30	**Session 3**			
	Architecture: Product Line Feature Models Session Moderator: Jan Bosch, University of Groningen		**Business: Panel** Session Moderator: Charles Krueger, BigLever Software	
	1:00 – 1:30	*A Methodology for the Derivation and Verification of Use Cases for Product Lines* A. Fantechi, Università di Firenze S. Gnesi, Istituto di Scienze e Tecnologie dell'Informazione "A.Faedo" G. Lami, Istituto di Scienze e Tecnologie dell'Informazione "A.Faedo" E. Nesti, Università di Firenze	1:00 – 2:30	*Avoiding, Surviving, and Prevailing Over Pitfalls in Product Line Engineering*
	1:30 – 2:00	*Staged Configuration Using Feature Models* Krzysztof Czarnecki, University of Waterloo Simon Helsen, University of Waterloo Ulrich Eisenecker, University of Applied Sciences Kaiserslautern		
	2:00 – 2:30	*Scenario-Based Decision Making for Architectural Variability in Product Families* Pierre America, Philips Research Dieter Hammer, Technical University Eindhoven Mugurel T. Ionita, Technical University Eindhoven Henk Obbink, Philips Research Eelco Rommes, Philips Research		
2:30 – 3:00	Refreshment Break			
3:00 – 4:00	**Session 4** **Product Line Hall of Fame** Session Moderator: Paul Clements, Software Engineering Institute			
4:00 – 4:15	**Wrap-Up and Future Plans** Linda Northrop, SPLC steering committee chair			
4:15	End of Conference			

The Third Software Product Line Conference

SPLC 2004

Conference Program Showing All Tracks

Track Indicators

- Architecture for Product Lines
- Business and Economic Issues
- Quality Assurance

Day 1 - Monday, August 30th, 2004

7:30 – 8:30	Continental Breakfast				
8:30 – 12:00	**Workshop:** Solutions for Automotive Software Architectures: Open Standards, References, and Product Line Architectures S. Ferber, A. Krüger, S. Voget, M. Weber	**Workshop:** Quality Assurance in Reuse Contexts R. Kolb, J. D. McGregor, and D. Muthig	**Tutorial:** An Introduction to Software Product Lines L. Northrop and P. Clements	**Workshop:** SPLYR - The First Software Product Lines Young Researchers Workshop B. Geppert, I. John, G. Lami	**Tutorial:** Architecture-Centric Software Engineering J. Bosch	**Tutorial:** Starting Product Lines (I) — Systematic Product Line Planning and Adoption K. Schmid and I. John
10:00 – 10:30	Refreshment Break					
12:00 – 1:30	Lunch					
1:30 – 5:00	**Workshop Continued**	**Workshop Continued**	**Tutorial Continued**	**Tutorial:** Developing a Measurement Program for Software Product Lines S. Cohen, D. Zubrow, G. Chastek	**Tutorial:** Software Variability Management J. Bosch	**Tutorial:** Starting Product Lines (II) — Product Line Analysis and Modeling I. John and K. Schmid
3:00 – 3:30	Refreshment Break					

Day 2 - Tuesday, August 31st, 2004

7:30 – 8:30	Continental Breakfast						
8:30 – 12:00	**Workshop:** SPLiT - Workshop on Software Product Line Testing B. Geppert, C. Krueger, J. Li	**Tutorial:** Product Line Analysis G. Chastek and P. Donohoe	**Tutorial:** Adopting Software Product Lines L. Northrop and L. Jones	**Workshop:** Software Variability Management for Product Derivation - Towards Tool Support T. Männistö and J. Bosch	**Workshop:** Modeling Business Issues of Software Product Lines S. Cohen	**Tutorial:** Product Line Architectures for Global Software Development D. J. Paulish, R. Pichler, and W. Kuhn	**Tutorial:** Quality Assurance for Software Product Lines R. Kolb and D. Muthig
10:00 – 10:30	Refreshment Break						
12:00 – 1:30	Lunch						
1:30 – 5:00	**Workshop Continued**	**Tutorial:** Industrial- Strength Software Product line Engineering J. Klein, D. Hill, and D. Weiss	**Tutorial:** Generative Software Development K. Czarnecki	**Workshop Continued**	**Workshop Continued**	**Tutorial:** Using Domain- Specific Languages, Patterns, Frameworks, and Tools to Assemble Applications J. Greenfield	**Tutorial:** Designing Software Product Lines with the Unified Modeling Language (UML) H. Gomaa
3:00 – 3:30	Refreshment Break						
5:30 – 7:30	**Social Reception**						

Day 3 - Wednesday, September 1st, 2004

8:00 – 8:30	Continental Breakfast		
8:30 – 10:00	**Session 1**		
	8:30 – 9:00	Welcome	
	9:00 – 10:00	**Keynote** *Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools* Jack Greenfield, Microsoft	
10:00 – 10:30	Refreshment Break		
10:30 – 12:00	**Session 2** **Business: Experiences in Introducing Product Line Engineering** Session Moderator: Martin Verlage, Market Maker Software AG		
	10:30 – 11:00	*Governing Software Product Lines and Reorganizations* Truman M. Jolley, Boeing Commercial Airplanes David J. Kasik, Boeing Commercial Airplanes Tammy R. Ben, Boeing Commercial Airplanes	
	11:00 – 11:30	*Software Product Line Support in Coremetrics OA2004* James Snyder, Coremetrics, Inc. Harry Lai, Coremetrics, Inc. Shirish Reddy, Coremetrics, Inc. Jimmy Wan, Coremetrics, Inc.	
	11:30 –12:00	*Introducing the Product Line Approach at Bosch Gasoline Systems: Experiences and Practices* Mirjam Steger, Robert Bosch GmbH Christian Tischer, Robert Bosch GmbH Birgit Boss, Robert Bosch GmbH Andreas Müller, Robert Bosch GmbH Oliver Pertler, Robert Bosch GmbH Wolfgang Stolz, Robert Bosch GmbH Stefan Ferber, Robert Bosch GmbH	
12:00 – 1:00	Lunch		
1:00 – 2:30	**Session 3**		
	Workshop Report Session Moderator: Klaus Schmid, Fraunhofer IESE		**Architecture: Panel** Session Moderator: Charles Krueger, BigLever Software
	1:00 – 2:30	Workshop organizers report on results	1:00 – 2:30 *Product Line Binding Times: What You Don't Know Can Hurt You*
2:30 – 3:00	Refreshment Break		
3:00 – 4:30	**Session 4** **Architecture: Product Creation, Part I** Session Moderator: Stefan Ferber, Robert Bosch		
	3:00 – 3:30	*Four Mechanisms for Adaptable Systems: A Meta-Level Approach to Building a Software Product Line* Claudia Fritsch, Robert Bosch GmbH Burkhardt Renz, University of Applied Sciences Gießen-Friedberg	
	3:30 – 4:00	*Automatic Generation of Program Families by Model Restrictions* Andrzej Wasowski, IT University of Copenhagen	
	4:00 – 4:30	*Software Product Lines in ArchJava* Sebastian Pavel, Ecole des Mines de Nantes - INRIA Jacques Noyé, Ecole des Mines de Nantes - INRIA Jean-Claude Royer, Ecole des Mines de Nantes - INRIA	
5:30 – 8:30	**Demonstrations**		
	5:30 & 7:00 RequiLine: A Requirements Engineering Tool for Software Product Lines Thomas Van Der Massen 6:00 & 7:30 MetaEdit+ metaCASE tool for Domain-Specific Modeling and Product Generation Juha-Pekka Tolvanen	5:30 & 7:00 ConIPF: Configuration in Industrial Product Families John MacGregor 6:00 & 7:30 Understanding Product Line Architectures Using Dependency Models Neeraj Sangal & Ev Jordan	5:30 & 7:30 Software Mass Customization with BigLever Software's GEARS Charles Krueger 6:00 & 7:00 Tools Supporting Domain-Specific Languages Jack Greenfield 6:30 & 8:00 Building the Buzz: www.softwareproductlines.com Charles Krueger
7:00 – 9:00	**Birds-of-a-Feather Sessions**		

Day 4 - Thursday, September 2nd, 2004

8:00 – 8:30	Continental Breakfast			
8:30 – 10:00	**Session 1**			
	Business: Frameworks for Introducing Product Line Engineering Session Moderator: Dirk Muthig, Fraunhofer IESE		**Quality Assurance: Panel** Session Moderator: Charles Krueger, BigLever Software	
	8:30 – 9:00	*Software Product Family Evaluation* Frank van der Linden, Philips Medical Systems Jan Bosch, University of Groningen Erik Kamsties, University of Duisburg-Essen Kari Känsälä, Nokia Research Center Henk Obbink, Philips Research Laboratories	8:30 – 10:00	*How Can Testing Keep Pace with Accelerated Development in Software Product Lines?*
	9:00 – 9:30	*Practical Evaluation of Software Product Family Architectures* Eila Niemelä, VTT Technical Research Centre of Finland Mari Matinlassi, VTT Technical Research Centre of Finland Anne Taulavuori, VTT Technical Research Centre of Finland		
	9:30 – 10:00	*On the Development of Software Product Family Components* Jan Bosch, University of Groningen		
10:00 – 10:30	Refreshment Break			
10:30 – 12:00	**Session 2**			
	Architecture: Product Creation, Part II Session Moderator: Scott Preece, Motorola		**Quality Assurance: Evaluation and Planning** Session Moderator: Frank van der Linden, Philips	
	10:30 – 11:00	*Experiences in Software Product Families: Problems and Issues During Product Derivation* Sybren Deelstra, University of Groningen Marco Sinnema, University of Groningen Jan Bosch, University of Groningen	10:30 – 11:00	*Observations from the Recovery of a Software Product Family* Patricia Lago, Vrije Universiteit Amsterdam Hans van Vliet, Vrije Universiteit Amsterdam
	11:00 – 11:30	*A Feature-Based Approach to Product Line Production Planning* Jaejoon Lee, Pohang University of Science and Technology Kyo C. Kang, Pohang University of Science and Technology Sajoong Kim, Korea Software Institute	11:00 – 11:30	*Product Line Potential Analysis* Claudia Fritsch, Robert Bosch GmbH Ralf Hahn, Robert Bosch GmbH

	11:30 – 12:00	*COVAMOF: A Framework for Modeling Variability in Software Product Families* Marco Sinnema, University of Groningen Sybren Deelstra, University of Groningen Jos Nijhuis, University of Groningen Jan Bosch, University of Groningen	11:30 – 12:00	*Generalized Release Planning for Product Line Architectures* Louis J. M. Taborda, Macquarie University
12:00 – 1:00	Lunch			
1:00 – 2:30	**Session 3**			
	Architecture: Product Line Feature Models Session Moderator: Jan Bosch, University of Groningen		**Business: Panel** Session Moderator: Charles Krueger, BigLever Software	
	1:00 – 1:30	*A Methodology for the Derivation and Verification of Use Cases for Product Lines* A. Fantechi, Università di Firenze S. Gnesi, Istituto di Scienze e Tecnologie dell'Informazione "A.Faedo" G. Lami, Istituto di Scienze e Tecnologie dell'Informazione "A.Faedo" E. Nesti, Università di Firenze	1:00 – 2:30	*Avoiding, Surviving, and Prevailing Over Pitfalls in Product Line Engineering*
	1:30 – 2:00	*Staged Configuration Using Feature Models* Krzysztof Czarnecki, University of Waterloo Simon Helsen, University of Waterloo Ulrich Eisenecker, University of Applied Sciences Kaiserslautern		
	2:00 – 2:30	*Scenario-Based Decision Making for Architectural Variability in Product Families* Pierre America, Philips Research Dieter Hammer, Technical University Eindhoven Mugurel T. Ionita, Technical University Eindhoven Henk Obbink, Philips Research Eelco Rommes, Philips Research		
2:30 – 3:00	Refreshment Break			
3:00 – 4:00	**Session 4** **Product Line Hall of Fame** Session Moderator: Paul Clements, Software Engineering Institute			
4:00 – 4:15	**Wrap-Up and Future Plans** Linda Northrop, SPLC steering committee chair			
4:15	End of Conference			

The Third Software Product Line Conference

SPLC 2004

Track Information (by day)

Day 1 - Monday, August 30th, 2004

Track: Architecture for Product Lines

8:30 – 12:00 **Tutorial:** Architecture-Centric Software Engineering
J. Bosch

1:30 – 5:00 **Tutorial:** Software Variability Management
J. Bosch

8:30 – 5:00 **Workshop:** Solutions for Automotive Software Architectures: Open Standards, References, and Product Line Architectures
S. Ferber, A. Krüger, S. Voget, M. Weber

Track: Business and Economic Issues

1:30 – 5:00 **Tutorial:** Developing a Measurement Program for Software Product Lines
S. Cohen, D. Zubrow, G. Chastek

1:30 – 5:00 **Tutorial:** Starting Product Lines (II) — Product Line Analysis and Modeling
I. John and K. Schmid

Track: Quality Assurance

1:30 – 5:00 **Tutorial:** Developing a Measurement Program for Software Product Lines
S. Cohen, D. Zubrow, G. Chastek

8:30 – 5:00 **Workshop:** Quality Assurance in Reuse Contexts
R. Kolb, J. D. McGregor, and D. Muthig

Topics Not Associated with any Tracks

8:30 – 5:00 **Tutorial:** An Introduction to Software Product Lines
L. Northrop and P. Clements

8:30 – 12:00 **Tutorial:** Starting Product Lines (I) — Systematic Product Line Planning and Adoption
K. Schmid and I. John

8:30 – 12:00 **Workshop:** SPLYR - The First Software Product Lines Young Researchers Workshop
B. Geppert, I. John, G. Lami

Day 2 - Tuesday, August 31st, 2004

Track: Architecture for Product Lines

8:30 – 12:00 **Tutorial:** Product Line Architectures for Global Software Development
D. J. Paulish, R. Pichler, and W. Kuhn

1:30 – 5:00 **Tutorial:** Generative Software Development
K. Czarnecki

1:30 – 5:00 **Tutorial:** Designing Software Product Lines with the Unified Modeling Language (UML)
H. Gomaa

1:30 – 5:00 **Tutorial:** Industrial-Strength Software Product line Engineering
J. Klein, D. Hill, and D. Weiss

1:30 – 5:00 **Tutorial:** Using Domain-Specific Languages, Patterns, Frameworks, and Tools to Assemble Applications
J. Greenfield

8:30 – 5:00 **Workshop:** Software Variability Management for Product Derivation - Towards Tool Support
T. Männistö and J. Bosch

Track: Business and Economic Issues

8:30 – 12:00 **Tutorial:** Product Line Analysis
G. Chastek and P. Donohoe

1:30 – 5:00 **Tutorial:** Industrial-Strength Software Product line Engineering
J. Klein, D. Hill, and D. Weiss

8:30 – 5:00 **Workshop:** Modeling Business Issues of Software Product Lines
S. Cohen

Track: Quality Assurance

8:30 – 12:00 **Tutorial:** Quality Assurance for Software Product Lines
R. Kolb and D. Muthig

8:30 – 5:00 **Workshop:** SPLiT - Workshop on Software Product Line Testing
B. Geppert, C. Krueger, J. Li

Topics Not Associated with any Tracks

8:30 – 12:00 **Tutorial:** Adopting Software Product Lines
L. Northrop and L. Jones

Day 3 - Wednesday, September 1st, 2004

Track: Architecture for Product Lines

1:00 – 2:30 **Panel:** *Product Line Binding Times: What You Don't Know Can Hurt You*

3:00 – 4:30 **Paper Session: Product Creation, Part I**

3:00 – 3:30 *Four Mechanisms for Adaptable Systems: A Meta-Level Approach to Building a Software Product Line*
Claudia Fritsch, Robert Bosch GmbH
Burkhardt Renz, University of Applied Sciences Gießen-Friedberg

3:30 – 4:00 *Automatic Generation of Program Families by Model Restrictions*
Andrzej Wasowski, IT University of Copenhagen

4:00 – 4:30 *Software Product Lines in ArchJava*
Sebastian Pavel, Ecole des Mines de Nantes - INRIA
Jacques Noyé, Ecole des Mines de Nantes - INRIA
Jean-Claude Royer, Ecole des Mines de Nantes - INRIA

Track: Business and Economic Issues

10:30 – 12:00 Paper Session: Experiences in Introducing Product Line Engineering

10:30 – 11:00 *Governing Software Product Lines and Reorganizations*
Truman M. Jolley, Boeing Commercial Airplanes
David J. Kasik, Boeing Commercial Airplanes
Tammy R. Ben, Boeing Commercial Airplanes

11:00 – 11:30 *Software Product Line Support in Coremetrics OA2004*
James Snyder, Coremetrics, Inc.
Harry Lai, Coremetrics, Inc.
Shirish Reddy, Coremetrics, Inc.
Jimmy Wan, Coremetrics, Inc.

11:30 –12:00 *Introducing the Product Line Approach at Bosch Gasoline Systems: Experiences and Practices*
Mirjam Steger, Robert Bosch GmbH
Christian Tischer, Robert Bosch GmbH
Birgit Boss, Robert Bosch GmbH
Andreas Müller, Robert Bosch GmbH
Oliver Pertler, Robert Bosch GmbH
Wolfgang Stolz, Robert Bosch GmbH
Stefan Ferber, Robert Bosch GmbH

Track: Quality Assurance

No papers or panels are scheduled for the quality assurance on day 3.

Topics Not Associated with any Tracks

9:00 – 10:00 **Keynote:** *Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools*
Jack Greenfield, Microsoft

1:00 – 2:30 **Workshop Report:** Workshop organizers report on results

5:30 – 8:30 **Demonstrations**

Day 4 - Thursday, September 2nd, 2004

Track: Architecture for Product Lines

10:30 – 12:00 Paper Session: Product Creation, Part II

10:30 – 11:00 *Experiences in Software Product Families: Problems and Issues During Product Derivation*
Sybren Deelstra, University of Groningen
Marco Sinnema, University of Groningen
Jan Bosch, University of Groningen

11:00 – 11:30 *A Feature-Based Approach to Product Line Production Planning*
Jaejoon Lee, Pohang University of Science and Technology
Kyo C. Kang, Pohang University of Science and Technology
Sajoong Kim, Korea Software Institute

11:30 – 12:00 *COVAMOF: A Framework for Modeling Variability in Software Product Families*
Marco Sinnema, University of Groningen
Sybren Deelstra, University of Groningen
Jos Nijhuis, University of Groningen
Jan Bosch, University of Groningen

1:00 – 2:30 Paper Session: Product Line Feature Models

1:00 – 1:30 *A Methodology for the Derivation and Verification of Use Cases for Product Lines*
A. Fantechi, Università di Firenze
S. Gnesi, Istituto di Scienze e Tecnologie dell'Informazione "A.Faedo"
G. Lami, Istituto di Scienze e Tecnologie dell'Informazione "A.Faedo"
E. Nesti, Università di Firenze

1:30 – 2:00 *Staged Configuration Using Feature Models*
Krzysztof Czarnecki, University of Waterloo
Simon Helsen, University of Waterloo
Ulrich Eisenecker, University of Applied Sciences Kaiserslautern

2:00 – 2:30 *Scenario-Based Decision Making for Architectural Variability in Product Families*
Pierre America, Philips Research
Dieter Hammer, Technical University Eindhoven
Mugurel T. Ionita, Technical University Eindhoven
Henk Obbink, Philips Research
Eelco Rommes, Philips Research

Track: Business and Economic Issues

8:30 – 10:00 Paper Session: Frameworks for Introducing Product Line Engineering

8:30 – 9:00 *Software Product Family Evaluation*
Frank van der Linden, Philips Medical Systems
Jan Bosch, University of Groningen
Erik Kamsties, University of Duisburg-Essen
Kari Känsälä, Nokia Research Center
Henk Obbink, Philips Research Laboratories

9:00 – 9:30 *Practical Evaluation of Software Product Family Architectures*
Eila Niemelä, VTT Technical Research Centre of Finland
Mari Matinlassi, VTT Technical Research Centre of Finland
Anne Taulavuori, VTT Technical Research Centre of Finland

9:30 – 10:00 *On the Development of Software Product Family Components*
Jan Bosch, University of Groningen

1:00 – 2:30 Panel: *Avoiding, Surviving, and Prevailing Over Pitfalls in Product Line Engineering*

Track: Quality Assurance

8:30 – 10:00 Panel: *How Can Testing Keep Pace with Accelerated Development in Software Product Lines?*

10:30 – 12:00 Paper Session: Evaluation and Planning

10:30 – 11:00 *Observations from the Recovery of a Software Product Family*
Patricia Lago, Vrije Universiteit Amsterdam
Hans van Vliet, Vrije Universiteit Amsterdam

11:00 – 11:30 *Product Line Potential Analysis*
Claudia Fritsch, Robert Bosch GmbH
Ralf Hahn, Robert Bosch GmbH

11:30 – 12:00 *Generalized Release Planning for Product Line Architectures*
Louis J. M. Taborda, Macquarie University

Topics Not Associated with any Tracks

3:00 – 4:00 Product Line Hall of Fame

4:00 – 4:15 Wrap-Up and Future Plans

The Third Software Product Line Conference

Track: Architecture for Product Lines

SPLC 2004

denotes topics in the Architecture for Product Lines track

Day 1 - Monday, August 30th, 2004

7:30 – 8:30	Continental Breakfast				
8:30 – 12:00	**Workshop:** Solutions for Automotive Software Architectures: Open Standards, References, and Product Line Architectures. S. Ferber, A. Krüger, S. Voget, M. Weber	**Workshop:** Quality Assurance in Reuse Contexts R. Kolb, J. D. McGregor, and D. Muthig	**Tutorial:** An Introduction to Software Product Lines L. Northrop and P. Clements	**Workshop:** SPLYR - The First Software Product Lines Young Researchers Workshop B. Geppert, I. John, G. Lami	**Tutorial:** Architecture-Centric Software Engineering J. Bosch	**Tutorial:** Starting Product Lines (I) — Systematic Product Line Planning and Adoption K. Schmid and I. John
10:00 – 10:30	Refreshment Break					
12:00 – 1:30	Lunch					
1:30 – 5:00	**Workshop Continued**	**Workshop Continued**	**Tutorial Continued**	**Tutorial:** Developing a Measurement Program for Software Product Lines S. Cohen, D. Zubrow, G. Chastek	**Tutorial:** Software Variability Management J. Bosch	**Tutorial:** Starting Product Lines (II) — Product Line Analysis and Modeling I. John and K. Schmid
3:00 – 3:30	Refreshment Break					

Day 2 - Tuesday, August 31st, 2004

7:30 – 8:30	Continental Breakfast					
8:30 – 12:00	**Workshop:** SPLIT - Workshop on Software Product Line Testing B. Geppert, C. Krueger, J. Li	**Tutorial:** Product Line Analysis G. Chastek and P. Donohoe	**Tutorial:** Adopting Software Product Lines L. Northrop and L. Jones	**Workshop:** Software Variability Management for Product Derivation - Towards Tool Support T. Männistö and J. Bosch	**Workshop:** Modeling Business Issues of Software Product Lines S. Cohen	**Tutorial:** Product Line Architectures for Global Software Development D. J. Paulish, R. Pichler, and W. Kuhn	**Tutorial:** Quality Assurance for Software Product Lines R. Kolb and D. Muthig
10:00 – 10:30	Refreshment Break						
12:00 – 1:30	Lunch						
1:30 – 5:00	**Workshop Continued**	**Tutorial:** Industrial Strength Software Product line Engineering J. Klein, D. Hill, and D. Weiss	**Tutorial:** Generative Software Development K. Czarnecki	**Workshop Continued**	**Workshop Continued**	**Tutorial:** Using Domain-Specific Languages, Patterns, Frameworks, and Tools to Assemble Applications J. Greenfield	**Tutorial:** Designing Software Product Lines with the Unified Modeling Language (UML) H. Gomaa
3:00 – 3:30	Refreshment Break						
5:30 – 7:30	**Social Reception**						

Day 3 - Wednesday, September 1st, 2004

8:00 – 8:30	Continental Breakfast		
8:30 – 10:00	**Session 1**		
	8:30 – 9:00	Welcome	
	9:00 – 10:00	**Keynote** *Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools* Jack Greenfield, Microsoft	
10:00 – 10:30	Refreshment Break		
10:30 – 12:00	**Session 2** **Business: Experiences in Introducing Product Line Engineering** Session Moderator: Martin Verlage, Market Maker Software AG		
	10:30 – 11:00	*Governing Software Product Lines and Reorganizations* Truman M. Jolley, Boeing Commercial Airplanes David J. Kasik, Boeing Commercial Airplanes Tammy R. Ben, Boeing Commercial Airplanes	
	11:00 – 11:30	*Software Product Line Support in Coremetrics OA2004* James Snyder, Coremetrics, Inc. Harry Lai, Coremetrics, Inc. Shirish Reddy, Coremetrics, Inc. Jimmy Wan, Coremetrics, Inc.	
	11:30 –12:00	*Introducing the Product Line Approach at Bosch Gasoline Systems: Experiences and Practices* Miriam Stieger, Robert Bosch GmbH Christian Tischer, Robert Bosch GmbH Birgit Boss, Robert Bosch GmbH Andreas Müller, Robert Bosch GmbH Oliver Pertler, Robert Bosch GmbH Wolfgang Stolz, Robert Bosch GmbH Stefan Ferber, Robert Bosch GmbH	
12:00 – 1:00	Lunch		
1:00 – 2:30	**Session 3**		
	Workshop Report Session Moderator: Klaus Schmid, Fraunhofer IESE		**Architecture: Panel** Session Moderator: Charles Krueger, BigLever Software
	1:00 – 2:30	Workshop organizers report on results	1:00 – 2:30 *Product Line Binding Times: What You Don't Know Can Hurt You*
2:30 – 3:00	Refreshment Break		
3:00 – 4:30	**Session 4** **Architecture: Product Creation, Part I** Session Moderator: Stefan Ferber, Robert Bosch		
	3:00 – 3:30	*Four Mechanisms for Adaptable Systems: A Meta-Level Approach to Building a Software Product Line* Claudia Fritsch, Robert Bosch GmbH Burkhardt Renz, University of Applied Sciences Gießen-Friedberg	
	3:30 – 4:00	*Automatic Generation of Program Families by Model Restrictions* Andrzej Wasowski, IT University of Copenhagen	
	4:00 – 4:30	*Software Product Lines in ArchJava* Sebastian Pavel, Ecole des Mines de Nantes - INRIA Jacques Noyé, Ecole des Mines de Nantes - INRIA Jean-Claude Royer, Ecole des Mines de Nantes - INRIA	
5:30 – 8:30	**Demonstrations**		
	5:30 & 7:00 RequLine: A Requirements Engineering Tool for Software Product Lines Thomas Van Der Massen 6:00 & 7:30 MetaEdit+ metaCASE tool for Domain-Specific Modeling and Product Generation Juha-Pekka Tolvanen	5:30 & 7:00 ConIPF: Configuration in Industrial Product Families John MacGregor 6:00 & 7:30 Understanding Product Line Architectures Using Dependency Models Neeraj Sangal & Ev Jordan	5:30 & 7:30 Software Mass Customization with BigLever Software's GEARS Charles Krueger 6:00 & 7:00 Tools Supporting Domain-Specific Languages Jack Greenfield 6:30 & 8:00 Building the Buzz: www.softwareproductlines.com Charles Krueger
7:00 – 9:00	**Birds-of-a-Feather Sessions**		

Day 4 - Thursday, September 2nd, 2004

8:00 – 8:30	Continental Breakfast			
8:30 – 10:00	**Session 1**			
	Business: Frameworks for Introducing Product Line Engineering Session Moderator: Dirk Muthig, Fraunhofer IESE		**Quality Assurance: Panel** Session Moderator: Charles Krueger, BigLever Software	
	8:30 – 9:00	*Software Product Family Evaluation* Frank van der Linden, Philips Medical Systems Jan Bosch, University of Groningen Erik Kamsties, University of Duisburg-Essen Kari Känsälä, Nokia Research Center Henk Obbink, Philips Research Laboratories	8:30 – 10:00	*How Can Testing Keep Pace with Accelerated Development in Software Product Lines?*
	9:00 – 9:30	*Practical Evaluation of Software Product Family Architectures* Eila Niemelä, VTT Technical Research Centre of Finland Mari Matinlassi, VTT Technical Research Centre of Finland Anne Taulavuori, VTT Technical Research Centre of Finland		
	9:30 – 10:00	*On the Development of Software Product Family Components* Jan Bosch, University of Groningen		
10:00 – 10:30	Refreshment Break			
10:30 – 12:00	**Session 2**			
	Architecture: Product Creation, Part II Session Moderator: Scott Preece, Motorola		**Quality Assurance: Evaluation and Planning** Session Moderator: Frank van der Linden, Philips	
	10:30 – 11:00	*Experiences in Software Product Families: Problems and Issues During Product Derivation* Sybren Deelstra, University of Groningen Marco Sinnema, University of Groningen Jan Bosch, University of Groningen	10:30 – 11:00	*Observations from the Recovery of a Software Product Family* Patricia Lago, Vrije Universiteit Amsterdam Hans van Vliet, Vrije Universiteit Amsterdam
	11:00 – 11:30	*A Feature-Based Approach to Product Line Production Planning* Jaejoon Lee, Pohang University of Science and Technology Kyo C. Kang, Pohang University of Science and Technology Sajoong Kim, Korea Software Institute	11:00 – 11:30	*Product Line Potential Analysis* Claudia Fritsch, Robert Bosch GmbH Ralf Hahn, Robert Bosch GmbH

	11:30 – 12:00	*COVAMOF: A Framework for Modeling Variability in Software Product Families* Marco Sinnema, University of Groningen Sybren Deelstra, University of Groningen Jos Nijhuis, University of Groningen Jan Bosch, University of Groningen	11:30 – 12:00	*Generalized Release Planning for Product Line Architectures* Louis J. M. Taborda, Macquarie University
12:00 – 1:00	Lunch			
1:00 – 2:30	**Session 3**			
	Architecture: Product Line Feature Models Session Moderator: Jan Bosch, University of Groningen		**Business: Panel** Session Moderator: Charles Krueger, BigLever Software	
	1:00 – 1:30	*A Methodology for the Derivation and Verification of Use Cases for Product Lines* A. Fantechi, Università di Firenze S. Gnesi, Istituto di Scienze e Tecnologie dell'Informazione "A.Faedo" G. Lami, Istituto di Scienze e Tecnologie dell'Informazione "A.Faedo" E. Nesti, Università di Firenze	1:00 – 2:30	*Avoiding, Surviving, and Prevailing Over Pitfalls in Product Line Engineering*
	1:30 – 2:00	*Staged Configuration Using Feature Models* Krzysztof Czarnecki, University of Waterloo Simon Helsen, University of Waterloo Ulrich Eisenecker, University of Applied Sciences Kaiserslautern		
	2:00 – 2:30	*Scenario-Based Decision Making for Architectural Variability in Product Families* Pierre America, Philips Research Dieter Hammer, Technical University Eindhoven Mugurel T. Ionita, Technical University Eindhoven Henk Obbink, Philips Research Eelco Rommes, Philips Research		
2:30 – 3:00	Refreshment Break			
3:00 – 4:00	**Session 4** **Product Line Hall of Fame** Session Moderator: Paul Clements, Software Engineering Institute			
4:00 – 4:15	**Wrap-Up and Future Plans** Linda Northrop, SPLC steering committee chair			
4:15	End of Conference			

The Third Software Product Line Conference

Track: Business and Economic Issues

SPLC 2004

denotes topics in the Business and Economic Issues track

Day 1 - Monday, August 30th, 2004

7:30 – 8:30	Continental Breakfast				
8:30 – 12:00	**Workshop:** Solutions for Automotive Software Architectures: Open Standards, References, and Product Line Architectures S. Ferber, A. Krüger, S. Voget, M. Weber	**Workshop:** Quality Assurance in Reuse Contexts R. Kolb, J. D. McGregor, and D. Muthig	**Tutorial:** An Introduction to Software Product Lines L. Northrop and P. Clements	**Workshop:** SPLYR - The First Software Product Lines Young Researchers Workshop B. Geppert, I. John, G. Lami	**Tutorial:** Architecture-Centric Software Engineering J. Bosch	**Tutorial:** Starting Product Lines (I) — Systematic Product Line Planning and Adoption K. Schmid and I. John
10:00 – 10:30	Refreshment Break					
12:00 – 1:30	Lunch					
1:30 – 5:00	**Workshop Continued**	**Workshop Continued**	**Tutorial Continued**	**Tutorial:** Developing a Measurement Program for Software Product Lines S. Cohen, D. Zubrow, G. Chastek	**Tutorial:** Software Variability Management J. Bosch	**Tutorial:** Starting Product Lines (II) — Product Line Analysis and Modeling I. John and K. Schmid
3:00 – 3:30	Refreshment Break					

Day 2 - Tuesday, August 31st, 2004

7:30 – 8:30	Continental Breakfast						
8:30 – 12:00	**Workshop:** SPLiT - Workshop on Software Product Line Testing B. Geppert, C. Krueger, J. Li	**Tutorial:** Product Line Analysis G. Chastek and P. Donohoe	**Tutorial:** Adopting Software Product Lines L. Northrop and L. Jones	**Workshop:** Software Variability Management for Product Derivation - Towards Tool Support T. Männistö and J. Bosch	**Workshop:** Modeling Business Issues of Software Product Lines S. Cohen	**Tutorial:** Product Line Architectures for Global Software Development D. J. Paulish, R. Pichler, and W. Kuhn	**Tutorial:** Quality Assurance for Software Product Lines R. Kolb and D. Muthig
10:00 – 10:30	Refreshment Break						
12:00 – 1:30	Lunch						
1:30 – 5:00	**Workshop Continued**	**Tutorial:** Industrial-Strength Software Product line Engineering J. Klein, D. Hill, and D. Weiss	**Tutorial:** Generative Software Development K. Czarnecki	**Workshop Continued**	**Workshop Continued**	**Tutorial:** Using Domain-Specific Languages, Patterns, Frameworks, and Tools to Assemble Applications J. Greenfield	**Tutorial:** Designing Software Product Lines with the Unified Modeling Language (UML) H. Gomaa
3:00 – 3:30	Refreshment Break						
5:30 – 7:30	**Social Reception**						

Day 3 - Wednesday, September 1st, 2004

8:00 – 8:30 Continental Breakfast

8:30 – 10:00 **Session 1**

8:30 – 9:00 Welcome

9:00 – 10:00 **Keynote**
Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools
Jack Greenfield, Microsoft

10:00 – 10:30 Refreshment Break

10:30 – 12:00 **Session 2**
Business: Experiences in Introducing Product Line Engineering
Session Moderator: Martin Verlage, Market Maker Software AG

10:30 – 11:00 *Governing Software Product Lines and Organizations*
Truman M. Jolley, Boeing Commercial Airplanes
David J. Kasik, Boeing Commercial Airplanes
Tammy R. Ben, Boeing Commercial Airplanes

11:00 – 11:30 *Software Product Line Support in Coremetrics OA2004*
James Snyder, Coremetrics, Inc.
Harry Lai, Coremetrics, Inc.
Shirish Reddy, Coremetrics, Inc.
Jimmy Wan, Coremetrics, Inc.

11:30 –12:00 *Introducing the Product Line Approach at Bosch Gasoline Systems: Experiences and Practices*
Mirjam Staeger, Robert Bosch GmbH
Christian Tischer, Robert Bosch GmbH
Birgit Boss, Robert Bosch GmbH
Andreas Müller, Robert Bosch GmbH
Oliver Pertler, Robert Bosch GmbH
Wolfgang Stolz, Robert Bosch GmbH
Stefan Ferber, Robert Bosch GmbH

12:00 – 1:00 Lunch

1:00 – 2:30 **Session 3**

Workshop Report
Session Moderator: Klaus Schmid, Fraunhofer IESE

1:00 – 2:30 Workshop organizers report on results

Architecture: Panel
Session Moderator: Charles Krueger, BigLever Software

1:00 – 2:30 *Product Line Binding Times: What You Don't Know Can Hurt You*

2:30 – 3:00 Refreshment Break

3:00 – 4:30 **Session 4**
Architecture: Product Creation, Part I
Session Moderator: Stefan Ferber, Robert Bosch

3:00 – 3:30 *Four Mechanisms for Adaptable Systems: A Meta-Level Approach to Building a Software Product Line*
Claudia Fritsch, Robert Bosch GmbH
Burkhardt Renz, University of Applied Sciences Gießen-Friedberg

3:30 – 4:00 *Automatic Generation of Program Families by Model Restrictions*
Andrzej Wasowski, IT University of Copenhagen

4:00 – 4:30 *Software Product Lines in AcmeJava*
Sebastian Pavel, Ecole des Mines de Nantes - INRIA
Jacques Noyé, Ecole des Mines de Nantes - INRIA
Jean-Claude Royer, Ecole des Mines de Nantes - INRIA

5:30 – 8:30 **Demonstrations**

5:30 & 7:00
ReqaLine: A Requirements Engineering Tool for Software Product Lines
Thomas Van Der Massen

6:00 & 7:30
MetaEdit+ metaCASE tool for Domain-Specific Modeling and Product Generation
Juha-Pekka Tolvanen

5:30 & 7:00
CompPF: Configuration in Industrial Product Families
John MacGregor

6:00 & 7:30
Understanding Product Line Architectures Using Dependency Models
Neeraj Sangal & Ev Jordan

5:30 & 7:30
Software Mass Customization with BigLever Software's GEARS
Charles Krueger

6:00 & 7:00
Tools Supporting Domain-Specific Languages
Jack Greenfield

6:30 & 8:00
Building the Buzz: www.softwareproductlines.com
Charles Krueger

7:00 – 9:00 **Birds-of-a-Feather Sessions**

Day 4 - Thursday, September 2nd, 2004

8:00 – 8:30 Continental Breakfast

8:30 – 10:00 **Session 1**

Business: Frameworks for Introducing Product Line Engineering
Session Moderator: Dirk Muthig, Fraunhofer IESE

8:30 – 9:00 *Software Product Family Evaluation*
Frank van der Linden, Philips Medical Systems
Jan Bosch, University of Groningen
Erik Kamsties, University of Dusburg-Essen
Kari Känsälä, Nokia Research Center
Henk Obbink, Philips Research Laboratories

9:00 – 9:30 *Practical Evaluation of Software Product Family Architectures*
Ella Niemelä, VTT Technical Research Centre of Finland
Mari Matinlassi, VTT Technical Research Centre of Finland
Anne Taulavuori, VTT Technical Research Centre of Finland

9:30 – 10:00 *On the Development of Software Product Family Components*
Jan Bosch, University of Groningen

Quality Assurance: Panel
Session Moderator: Charles Krueger, BigLever Software

8:30 – 10:00 *How Can Testing Keep Pace with Accelerated Development in Software Product Lines?*

10:00 – 10:30 Refreshment Break

10:30 – 12:00 **Session 2**

Architecture: Product Creation, Part II
Session Moderator: Scott Preece, Motorola

10:30 – 11:00 *Experiences in Software Product Families: Problems and Issues During Product Derivation*
Sybren Deelstra, University of Groningen
Marco Sinnema, University of Groningen
Jan Bosch, University of Groningen

11:00 – 11:30 *A Feature-Based Approach to Product Line Production Planning*
Jaejoon Lee, Pohang University of Science and Technology
Kyo C. Kang, Pohang University of Science and Technology
Sajoong Kim, Korea Software Institute

Quality Assurance: Evaluation and Planning
Session Moderator: Frank van der Linden, Philips

10:30 – 11:00 *Observations from the Recovery of a Software Product Family*
Patricia Lago, Vrije Universiteit Amsterdam
Hans van Vliet, Vrije Universiteit Amsterdam

11:00 – 11:30 *Product Line Potential Analysis*
Claudia Fritsch, Robert Bosch GmbH
Ralf Hahn, Robert Bosch GmbH

	11:30 – 12:00	*COVAMOF: A Framework for Modeling Variability in Software Product Families* Marco Sinnema, University of Groningen Sybren Deelstra, University of Groningen Jos Nijhuis, University of Groningen Jan Bosch, University of Groningen	11:30 – 12:00	*Generalized Release Planning for Product Line Architectures* Louis J. M. Taborda, Macquarie University
12:00 – 1:00	Lunch			
1:00 – 2:30	**Session 3**			
	Architecture: Product Line Feature Models Session Moderator: Jan Bosch, University of Groningen		**Business: Panel** Session Moderator: Charles Krueger, BigLever Software	
	1:00 – 1:30	*A Methodology for the Derivation and Verification of Use Cases for Product Lines* A. Fantechi, Università di Firenze S. Gnesi, Istituto di Scienze e Tecnologie dell'Informazione "A.Faedo" G. Lami, Istituto di Scienze e Tecnologie dell'Informazione "A.Faedo" E. Nesti, Università di Firenze	1:00 – 2:30	*Avoiding, Surviving, and Prevailing Over Pitfalls in Product Line Engineering*
	1:30 – 2:00	*Staged Configuration Using Feature Models* Krzysztof Czarnecki, University of Waterloo Simon Helsen, University of Waterloo Ulrich Eisenecker, University of Applied Sciences Kaiserslautern		
	2:00 – 2:30	*Scenario-Based Decision Making for Architectural Variability in Product Families* Pierre America, Philips Research Dieter Hammer, Technical University Eindhoven Mugurel T. Ionita, Technical University Eindhoven Henk Obbink, Philips Research Eelco Rommes, Philips Research		
2:30 – 3:00	Refreshment Break			
3:00 – 4:00	**Session 4** **Product Line Hall of Fame** Session Moderator: Paul Clements, Software Engineering Institute			
4:00 – 4:15	**Wrap-Up and Future Plans** Linda Northrop, SPLC steering committee chair			
4:15	End of Conference			

The Third Software Product Line Conference

Track: Quality Assurance

SPLC 2004

denotes topics in the Quality Assurance track

Day 1 - Monday, August 30th, 2004

7:30 – 8:30 Continental Breakfast

8:30 – 12:00

Workshop: Solutions for Automotive Software Architectures: Open Standards, References, and Product Line Architectures
S. Ferber, A. Krüger, S. Voget, M. Weber

Workshop: Quality Assurance in Reuse Contexts
R. Kolb, J. D. McGregor, and D. Muthig

Tutorial: An Introduction to Software Product Lines
L. Northrop and P. Clements

Workshop: SPLYR - The First Software Product Lines Young Researchers Workshop
B. Geppert, I. John, G. Lami

Tutorial: Architecture-Centric Software Engineering
J. Bosch

Tutorial: Starting Product Lines (I) — Systematic Product Line Planning and Adoption
K. Schmid and I. John

10:00 – 10:30 Refreshment Break

12:00 – 1:30 Lunch

1:30 – 5:00

Workshop Continued

Workshop Continued

Tutorial Continued

Tutorial: Developing a Measurement Program for Software Product Lines
S. Cohen, D. Zubrow, G. Chastek

Tutorial: Software Variability Management
J. Bosch

Tutorial: Starting Product Lines (II) — Product Line Analysis and Modeling
I. John and K. Schmid

3:00 – 3:30 Refreshment Break

Day 2 - Tuesday, August 31st, 2004

7:30 – 8:30 Continental Breakfast

8:30 – 12:00

Workshop: SPLiT - Workshop on Software Product Line Testing
B. Geppert, C. Krueger, J. Li

Tutorial: Product Line Analysis
G. Chastek and P. Donohoe

Tutorial: Adopting Software Product Lines
L. Northrop and L. Jones

Workshop: Software Variability Management for Product Derivation - Towards Tool Support
T. Männistö and J. Bosch

Workshop: Modeling Business Issues of Software Product Lines
S. Cohen

Tutorial: Product Line Architectures for Global Software Development
D. J. Paulish, R. Pichler, and W. Kuhn

Tutorial: Quality Assurance for Software Product Lines
R. Kolb and D. Muthig

10:00 – 10:30 Refreshment Break

12:00 – 1:30 Lunch

1:30 – 5:00

Workshop Continued

Tutorial: Industrial-Strength Software Product line Engineering
J. Klein, D. Hill, and D. Weiss

Tutorial: Generative Software Development
K. Czarnecki

Workshop Continued

Workshop Continued

Tutorial: Using Domain-Specific Languages, Patterns, Frameworks, and Tools to Assemble Applications
J. Greenfield

Tutorial: Designing Software Product Lines with the Unified Modeling Language (UML)
H. Gomaa

3:00 – 3:30 Refreshment Break

5:30 – 7:30 **Social Reception**

Day 3 - Wednesday, September 1st, 2004

8:00 – 8:30 Continental Breakfast

8:30 – 10:00 **Session 1**

8:30 – 9:00 Welcome

9:00 – 10:00 **Keynote**
Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools
Jack Greenfield, Microsoft

10:00 – 10:30 Refreshment Break

10:30 – 12:00 **Session 2**
Business: Experiences in Introducing Product Line Engineering
Session Moderator: Martin Verlage, Market Maker Software AG

10:30 – 11:00 *Governing Software Product Lines and Reorganizations*
Truman M. Jolley, Boeing Commercial Airplanes
David J. Kasik, Boeing Commercial Airplanes
Tammy R. Ben, Boeing Commercial Airplanes

11:00 – 11:30 *Software Product Line Support in Coremetrics OA2004*
James Snyder, Coremetrics, Inc.
Harry Lai, Coremetrics, Inc.
Shirish Reddy, Coremetrics, Inc.
Jimmy Wan, Coremetrics, Inc.

11:30 –12:00 *Introducing the Product Line Approach at Bosch Gasoline Systems: Experiences and Practices*
Mirjam Staeger, Robert Bosch GmbH
Christian Tischer, Robert Bosch GmbH
Birgit Boss, Robert Bosch GmbH
Andreas Müller, Robert Bosch GmbH
Oliver Pertler, Robert Bosch GmbH
Wolfgang Stolz, Robert Bosch GmbH
Stefan Ferber, Robert Bosch GmbH

12:00 – 1:00 Lunch

1:00 – 2:30 **Session 3**

Workshop Report
Session Moderator: Klaus Schmid, Fraunhofer IESE

1:00 – 2:30 Workshop organizers report on results

Architecture: Panel
Session Moderator: Charles Krueger, BigLever Software

1:00 – 2:30 *Product Line Binding Times: What You Don't Know Can Hurt You*

2:30 – 3:00 Refreshment Break

3:00 – 4:30 **Session 4**
Architecture: Product Creation, Part I
Session Moderator: Stefan Ferber, Robert Bosch

3:00 – 3:30 *Four Mechanisms for Adaptable Systems: A Meta-Level Approach to Building a Software Product Line*
Claudia Fritsch, Robert Bosch GmbH
Burkhardt Renz, University of Applied Sciences Gießen-Friedberg

3:30 – 4:00 *Automatic Generation of Program Families by Model Restrictions*
Andrzej Wasowski, IT University of Copenhagen

4:00 – 4:30 *Software Product Lines in AcmJava*
Sebastian Pavel, Ecole des Mines de Nantes - INRIA
Jacques Noyé, Ecole des Mines de Nantes - INRIA
Jean-Claude Royer, Ecole des Mines de Nantes - INRIA

5:30 – 8:30 **Demonstrations**

5:30 & 7:00
ReqaLine: A Requirements Engineering Tool for Software Product Lines
Thomas Van Der Massen

6:00 & 7:30
MetaEdit+ metaCASE tool for Domain-Specific Modeling and Product Generation
Juha-Pekka Tolvanen

5:30 & 7:00
ConIPF: Configuration in Industrial Product Families
John MacGregor

6:00 & 7:30
Understanding Product Line Architectures Using Dependency Models
Neeraj Sangal & Ev Jordan

5:30 & 7:30
Software Mass Customization with BigLever Software's GEARS
Charles Krueger

6:00 & 7:00
Tools Supporting Domain-Specific Languages
Jack Greenfield

6:30 & 8:00
Building the Buzz: www.softwareproductlines.com
Charles Krueger

7:00 – 9:00 **Birds-of-a-Feather Sessions**

Day 4 - Thursday, September 2nd, 2004

8:00 – 8:30 Continental Breakfast

8:30 – 10:00 **Session 1**

Business: Frameworks for Introducing Product Line Engineering
Session Moderator: Dirk Muthig, Fraunhofer IESE

8:30 – 9:00 *Software Product Family Evaluation*
Frank van der Linden, Philips Medical Systems
Jan Bosch, University of Groningen
Erik Kamsties, University of Duisburg-Essen
Kari Känsälä, Nokia Research Center
Henk Obbink, Philips Research Laboratories

9:00 – 9:30 *Practical Evaluation of Software Product Family Architectures*
Eila Niemelä, VTT Technical Research Centre of Finland
Mari Matinlassi, VTT Technical Research Centre of Finland
Anne Taulavuori, VTT Technical Research Centre of Finland

9:30 – 10:00 *On the Development of Software Product Family Components*
Jan Bosch, University of Groningen

Quality Assurance: Panel
Session Moderator: Charles Krueger, BigLever Software

8:30 – 10:00 *How Can Testing Keep Pace with Accelerated Development in Software Product Lines?*

10:00 – 10:30 Refreshment Break

10:30 – 12:00 **Session 2**

Architecture: Product Creation, Part II
Session Moderator: Scott Preece, Motorola

10:30 – 11:00 *Experiences in Software Product Families: Problems and Issues During Product Derivation*
Sybren Deelstra, University of Groningen
Marco Sinnema, University of Groningen
Jan Bosch, University of Groningen

11:00 – 11:30 *A Feature-Based Approach to Product Line Production Planning*
Jaejoon Lee, Pohang University of Science and Technology
Kyo C. Kang, Pohang University of Science and Technology
Sajoong Kim, Korea Software Institute

Quality Assurance: Evaluation and Planning
Session Moderator: Frank van der Linden, Philips

10:30 – 11:00 *Observations from the Recovery of a Software Product Family*
Patricia Lago, Vrije Universiteit Amsterdam
Hans van Vliet, Vrije Universiteit Amsterdam

11:00 – 11:30 *Product Line Potential Analysis*
Claudia Fritsch, Robert Bosch GmbH
Ralf Hahn, Robert Bosch GmbH

	11:30 – 12:00	*COVAMOF: A Framework for Modeling Variability in Software Product Families* Marco Sinnema, University of Groningen Sybren Deelstra, University of Groningen Jos Nijhuis, University of Groningen Jan Bosch, University of Groningen	**11:30 – 12:00**	*Generalized Release Planning for Product Line Architectures* Louis J. M. Taborda, Macquarie University
12:00 – 1:00	Lunch			
1:00 – 2:30	**Session 3**			
	Architecture: Product Line Feature Models Session Moderator: Jan Bosch, University of Groningen		**Business: Panel** Session Moderator: Charles Krueger, BigLever Software	
	1:00 – 1:30	*A Methodology for the Derivation and Verification of Use Cases for Product Lines* A. Fantechi, Università di Firenze S. Gnesi, Istituto di Scienze e Tecnologie dell'Informazione "A.Faedo" G. Lami, Istituto di Scienze e Tecnologie dell'Informazione "A.Faedo" E. Nesti, Università di Firenze	1:00 – 2:30	*Avoiding, Surviving, and Prevailing Over Pitfalls in Product Line Engineering*
	1:30 – 2:00	*Staged Configuration Using Feature Models* Krzysztof Czarnecki, University of Waterloo Simon Helsen, University of Waterloo Ulrich Eisenecker, University of Applied Sciences Kaiserslautern		
	2:00 – 2:30	*Scenario-Based Decision Making for Architectural Variability in Product Families* Pierre America, Philips Research Dieter Hammer, Technical University Eindhoven Mugurel T. Ionita, Technical University Eindhoven Henk Obbink, Philips Research Eelco Rommes, Philips Research		
2:30 – 3:00	Refreshment Break			
3:00 – 4:00	**Session 4** **Product Line Hall of Fame** Session Moderator: Paul Clements, Software Engineering Institute			
4:00 – 4:15	**Wrap-Up and Future Plans** Linda Northrop, SPLC steering committee chair			
4:15	End of Conference			