The Second Software Product Line Conference

August 19–22, 2002
San Diego, California

The Software Engineering Institute (SEI) held the second Software Product Line Conference in San Diego, CA, August 19-22, 2002. There were 157 attendees, roughly two thirds from the United States and the remaining third from 16 other countries spanning North America, Europe, Asia, and Africa. Most of the participants were from commercial organizations, but academia and government were well represented. There was participation from most of the software product line leaders including: Hewlett Packard, Nokia, Philips, Robert Bosch GmBh, Avaya, Motorola, Cummins Inc., Siemens, Ericsson, Thales, and General Motors.

The conference program included seven tutorials; seven workshops, including one that was focused on DoD product line practice; an inspiring keynote talk, "Global Software Product Lines and Infinite Diversity," by Anders Heie from Nokia Mobile Phones; two panels; twenty-four technical paper presentations; four technical demonstrations; several Birds-of-a-Feather Sessions; and the Second Software Product Line Hall of Fame at which attendees voted in five new members to the software product line elite.

The Third Software Product Line Conference (SPLC3) will be held in the United States in autumn 2004. The Fifth Product Family Engineering Workshop will be held in Italy in autumn 2003. A mixture of SEI and international product line leaders will organize each.

Post-Conference Information

SPLC2 Hall of Fame Inductees

Slide Presentations

- **Keynote:** Global Software Product Lines and Infinite Diversity – Anders Heie
- **Crossing the Chasm Panel** – Scott E. Preece
- **Product Line Adoption and Institutionalization** – Günter Böckle, et. al.
- **Governance Polarities of Internal Product Lines** – Truman M. Jolley, et. al.
Conference Information

- Technical Program
- Tutorials
- Workshops
- Panels
- Product Line Hall of Fame
- Keynote Address
- Demonstrations
- Program Committee
- San Diego
- Registration Information

All conference activities were held at the Sheraton San Diego Hotel and Marina.
SPLC2 Hall of Fame Inductees

The following product lines have been inducted into the Software Product Line Hall of Fame. Nominations were accepted at the SPLC2 and participants voted on the nominees based on preestablished criteria for election.

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 six, Cummins changed the way they developed software and embraced the product line approach. Their product line is a story of 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 1000 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 an ROI of 10:1, from the product line approach. It has also enabled them 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.


### Telecommunication Switching System, Philips

The PKI tss (Telecommunications Switching System) 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 where 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 it could be built and tested incrementally. The family was very successful in having a fast time-to-market, and high reuse.

In 1994 PKI was sold to Lucent, which 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.


**5ESS telecommunications switch, Bell Labs / AT&T / Lucent**

The 5ESS™ 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 was 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 10MLOC. 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 successfully accomplished in the 5ESS software and documented in the software engineering literature. Domains such as switch maintenance, signalling, and traffic management showed productivity improvements of factors of 3 to 5 as a result.


**Bold Stroke avionics software family, Boeing**

The Bold Stroke Software Product Line is comprised of a wide range of artifacts required to create Operational Flight Programs for a variety of Boeing military fighters, including a highly configurable architecture, application components, middleware framework, and development processes and tools. Operational Flight Programs 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 an open source real-time Object Request Broker - The ACE ORB-developed in partnership with Washington University in St. Louis and a commercial real-time operating system. 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.


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 their 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, which they called MERGER. The result is a 520 KSLOC system that meets all of 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 their systematic product line approach, MARKET MAKER is able to 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 of their small, efficient team required for maintaining the running systems.

SPLC2 - Product Line Hall of Fame


SPLC1 Hall of Fame Inductees

The founding members of the SPLC Software Product Line Hall of Fame, inducted at SPLC1, are:

- **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 field of both software engineering and of product line engineering.

- **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 upon a robust architecture that was designed to handle both of those initial systems as well as the more than fifty 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 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.
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 "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 and component management, task scheduling and event control. Nokia Mobile Phones is the world's largest mobile phone manufacturer, and they believe that software product line engineering has helped it to reach that position.

Owen Firmware Cooperative, Hewlett Packard
Owen is a community of firmware development teams from 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, firmware architect, 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 coop. 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.

Criteria for Election To 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, i.e., there is a way to tell whether or not a software system is a member of the product line, either by applying a known rule or a known enumeration.
- The family that constitutes the product line is explicitly defined and designed as a product line, i.e., 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 in creating their product-lines or in 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.
Global Software Product Lines and Infinite Diversity

SPLC2
San Diego, August 21st

Anders Heie
Nokia Mobile Phones
Governance Polarities of Internal Product Lines

Truman M. Jolley
David J. Kasik
Conrad E. Kimball
Boeing Commercial Airplanes

9/10/2002
Software Product Line Conference 2002

©2002, The Boeing Company
Wednesday, August 21, 2002

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>8:30 a.m.</td>
<td>Welcome</td>
</tr>
<tr>
<td>9:00 a.m.</td>
<td><strong>Keynote Speaker:</strong> Anders Heie, Nokia Mobile Phones</td>
</tr>
<tr>
<td>10:00 a.m.</td>
<td>Break</td>
</tr>
<tr>
<td>10:30 a.m.</td>
<td>Technical Session 1 - Variability</td>
</tr>
<tr>
<td></td>
<td>Moderator: Felix Bachmann, Software</td>
</tr>
<tr>
<td></td>
<td>Engineering Institute</td>
</tr>
<tr>
<td></td>
<td>On the Influence of Variabilities on the</td>
</tr>
<tr>
<td></td>
<td>Application Engineering Process of a</td>
</tr>
<tr>
<td></td>
<td>Product Family</td>
</tr>
<tr>
<td></td>
<td>Lars Geyer, University of Kaiserslautern</td>
</tr>
<tr>
<td></td>
<td>Martin Becker, University of Kaiserslautern</td>
</tr>
<tr>
<td>10:30 a.m.</td>
<td>Technical Session 2 - Organization</td>
</tr>
<tr>
<td></td>
<td>Issues I</td>
</tr>
<tr>
<td></td>
<td>Moderator: Paul Clements, Software</td>
</tr>
<tr>
<td></td>
<td>Engineering Institute</td>
</tr>
<tr>
<td></td>
<td>Adopting and Institutionalizing a Product</td>
</tr>
<tr>
<td></td>
<td>Line Culture</td>
</tr>
<tr>
<td></td>
<td>Günter W. Böckle, Siemens AG</td>
</tr>
<tr>
<td></td>
<td>Jesús Bermejo, Telvent</td>
</tr>
<tr>
<td></td>
<td>Peter Knauber, Fraunhofer IESE</td>
</tr>
<tr>
<td></td>
<td>Charles Krueger, BigLever Software, Inc.</td>
</tr>
<tr>
<td></td>
<td>Julio Cesar Leite, Pontifícia Universidade</td>
</tr>
<tr>
<td></td>
<td>Católica do Rio de Janeiro</td>
</tr>
<tr>
<td></td>
<td>Frank van der Linden, Philips</td>
</tr>
<tr>
<td>Time</td>
<td>Session</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>12:00 noon</td>
<td>Lunch</td>
</tr>
</tbody>
</table>
| 1:00 p.m.    | **Panel - Workshop Summaries**  
Moderator: Sholom Cohen, Software Engineering Institute |
| 2:15 p.m.    | Break                                                                                         |
| 2:30 p.m.    | **Technical Session 3 - Architecture Design**  
Moderator: Robert Nord, Siemens Research, Inc. |
|              | **Product Line Architecture and the Separation of Concerns**  
Jay van Zyl, Rubico Products (Pty) Ltd |
|              | **Adaptable Components for Software Product-Line Engineering**  
John Brown, Queen's University of Belfast  
Ivor Spence, Queen's University of Belfast  
Peter Kilpatrick, Queen's University of Belfast  
Danny Crookes, Queen's University of Belfast |
|              | **Model-driven Product Line Architectures**  
Dirk Muthig, Fraunhofer IESE  
Colin Atkinson, Fraunhofer IESE |
|              | **Using First Order Logic for Product Line Model Validation**  
Mike Mannion, Glasgow Caledonian University |
|              | **Technical Session 4 - Implementation**  
Moderator: Grady Campbell, Prosperity Heights Software |
|              | **Establishing a Software Product Line in an Immature Domain**  
Stefan Voget, Robert Bosch GmbH  
Martin Becker, University of Kaiserslautern |
|              | **Critical Factors for a Successful Platform-based Product Family Approach**  
Jan Gerben Wijnstra, Philips Research Laboratories |

**Representing Variability in Software Product Lines: A Case Study**  
Michel Jaring, University of Groningen  
Jan Bosch, University of Groningen

**Variation Management for Software Production Lines**  
Charles Krueger, BigLever Software, Inc.
Systematic Integration of Variability into Product Line Architecture Design

Steffen Thiel, Robert Bosch GmbH
Andreas Hein, Robert Bosch GmbH

Product line annotations with UML-F

Wolfgang Pree, University of California, Berkeley
Marcus Fontoura, IBM Almaden Research
Bernhard Rumpe, Munich University of Technology

4:00 p.m. Break

4:30 p.m. Panel: Tool Support for Product Lines: What We Have and What We Need

Moderator:
David Weiss, Director of Software Technology Research, Avaya Laboratories

Panelists:
Charles Krueger, BigLever Software, Inc.
Bedir Tekinerdogan, University of Twente
Grady Campbell, Prosperity Heights Software
Daniel Simon, Universität Stuttgart, Institut für Informatik

5:30 p.m. Presentations end. Birds of a feather sessions and demonstrations begin.

Thursday, August 22, 2002

9:00 a.m. Technical Session 5 - Feature Modeling
Moderator: Frank van der Linden, Philips Medical Systems

Feature Modeling: a Meta-Model to Enhance Usability and Usefulness
Daniel Fey, Nokia Research Center
Robert Fajta, Nokia Research Center
Andras Boros, Nokia Research Center

Assembling Software Products from Feature Selections
Arie van Deursen, CWI
Merin de Jonge, CWI
Tobias Kuipers, Software Improvement Group

Technical Session 6 - Organization Issue II
Moderator: Günter Böckle, Siemens AG

Maturity and Evolution in Software Product Lines: Approaches, Artifacts and Organization
Jan Bosch, University of Groningen

Evolutionary Introduction of Software Product Lines
Daniel Simon, Universität Stuttgart
Thomas Eisenbarth, Universität Stuttgart
<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:30 a.m.</td>
<td>Break</td>
</tr>
<tr>
<td>11:00 a.m.</td>
<td><strong>Panel:</strong> Software Product Lines: Crossing the Chasm</td>
</tr>
<tr>
<td></td>
<td>Moderator: Stuart Faulk, Department of Computer and Information Science, University of Oregon</td>
</tr>
<tr>
<td></td>
<td>Panelists: Dan Paulish, Siemens</td>
</tr>
<tr>
<td></td>
<td>Scott Preece, Motorola</td>
</tr>
<tr>
<td></td>
<td>Sergio Bandinelli, European Software Institute</td>
</tr>
<tr>
<td></td>
<td>Linda Northrop, Software Engineering Institute</td>
</tr>
<tr>
<td>12:00 p.m.</td>
<td>Lunch</td>
</tr>
<tr>
<td>1:00 p.m.</td>
<td><strong>Technical Session 7 - Components</strong></td>
</tr>
<tr>
<td></td>
<td>Moderator: Charles Krueger, BigLever Software, Inc.</td>
</tr>
<tr>
<td></td>
<td>Performance Analysis of Component-based Applications</td>
</tr>
<tr>
<td></td>
<td>Sherif Yacoub, Hewlett-Packard Laboratories</td>
</tr>
<tr>
<td></td>
<td><strong>Technical Session 8 - Choosing the Assets</strong></td>
</tr>
<tr>
<td></td>
<td>Moderator: Steffen Thiel, Robert Bosch GmbH</td>
</tr>
<tr>
<td></td>
<td>Method for Product-Line Scoping based on Decision-Making Framework</td>
</tr>
<tr>
<td></td>
<td>Tomoji Kishi, NEC Corporation</td>
</tr>
<tr>
<td></td>
<td>Natsuko Noda, NEC Corporation</td>
</tr>
<tr>
<td></td>
<td>Takuya Katayama, Japan Advanced Institute of Science and Technology</td>
</tr>
<tr>
<td>Time</td>
<td>Session</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------------------</td>
</tr>
<tr>
<td>2:30 p.m.</td>
<td>Break</td>
</tr>
<tr>
<td>3:00 p.m.</td>
<td><strong>Panel: Product Line Hall of Fame</strong></td>
</tr>
<tr>
<td></td>
<td>Moderator: Paul Clements, Software Engineering Institute</td>
</tr>
<tr>
<td>4:00 p.m.</td>
<td>Conference Ends</td>
</tr>
</tbody>
</table>

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

Copyright ©2002 by Carnegie Mellon University
This material is being posted by Carnegie Mellon University's Software Engineering Institute (SEI) on this Web site as a community service.

The Second Software Product Line Conference

Conference Tutorials

Monday, August 19

- Tutorial 1: Software Product Lines: Practices and Patterns (8:30 a.m.– 5:00 p.m.)
- Tutorial 2: Experiencing Product Line Adoption (8:30 a.m.– 12:00 p.m.)
- Tutorial 3: Building Reusable Test Assets for a Product Line (1:30 p.m.– 5:00 p.m.)

Tuesday, August 20

- Tutorial 4: Practical Product Line Scoping and Modeling (8:30 a.m.– 12:00 p.m.)
- Tutorial 5: Component-based Product-line Engineering with the UML (1:30 p.m.– 5:00 p.m.)
- Tutorial 6: Global Analysis: Developing Software Architecture Design Strategies to Respond to Variation and Anticipated Change (8:30 a.m.– 12:00 p.m.)
- Tutorial 7: Architecture-Centric Software Engineering (1:30 p.m.– 5:00 p.m.)

Tutorial 1: Software Product Lines: Practices and Patterns

Presenters: Linda Northrop and Paul Clements, Software Engineering Institute

Monday, August 19

Time: 8:30 a.m.– 5:00 p.m.
Room: Coronado A

Abstract: The primary aim of this tutorial is to help a decision-maker understand the basics of software product line strategies, the advantages of the approach, and what will be involved in adopting a product line approach. Additionally, the aim is to help them assess where their organization stands in relation to the capabilities needed to launch a product line and to give them the necessary tools to tailor adoption
and execution strategies to their organization.

**Intended Audience:** The tutorial is aimed at those in an organization who are in a position to make decisions affecting the decision to adopt a product line approach, and those in a position to carry out that decision. This includes technical managers at all levels, as well as those on the software development staff. Anyone who can act as a technology change agent will benefit from this tutorial.

---

**Tutorial 2: Experiencing Product Line Adoption**

**Presenter:** Grady Campbell, Prosperity Heights Software

**Monday, August 19**

**Time:** 8:30 a.m. – 12:00 p.m.

**Room:** Coronado B

**Abstract:** The goal of this tutorial is to give participants a roadmap for adopting a product line approach to software development. Guided by the reuse-driven Process Improvement (PIr) method, the tutorial will step participants through the activities of a systematic adoption process. Participants will be presented guidance on identifying market-oriented business objectives and success criteria, evaluating the viability of a product line approach for meeting those needs, determining related process improvement goals, and deriving a suitable strategy and plan for instituting product line practices.

**Intended Audience:** This tutorial is targeted to experienced engineers and managers whose organizations have a need to build similar products and, in doing so, want to increase their productivity and product quality. Tutorial attendees should be familiar with the principles and use of a conventional process improvement method such as the SEI Capability Maturity Model² approach.

---

**Tutorial 3: Building Reusable Test Assets for a Product Line**

**Presenter:** John D. McGregor, Clemson University

**Monday, August 19**

**Time:** 1:30 p.m. – 5:00 p.m.

**Room:** Coronado B

**Abstract:** The resources needed to test a software product can equal or exceed the resources required to create the product. This tutorial takes a typical comprehensive testing process that includes both "development" testing and "system" testing and identifies those aspects of software product line practice that can be applied to the development of test assets. Achieving profitable reuse requires the same disciplined, managed approach to the production of test assets as is applied to the creation of development assets. Creating a software architecture for the test software, using abstract and generic classes, and mirroring the architecture of the application in the test architecture are some of the
Intended Audience: This tutorial presents a comprehensive test process that includes roles for software developers, traditional testers and managers of both. The test process is tightly integrated with the development process during the early phases of development and becomes more independent as development proceeds. The tutorial will describe techniques used by each of the roles at each of the stages of development.

Tutorial 4: Practical Product Line Scoping and Modeling
Presenter: Isabel John, Fraunhofer IESE

Tuesday, August 20
Time: 8:30 a.m. – 12:00 p.m.
Room: Coronado A

Abstract: While the technologies for scoping and modeling of product lines are typically treated as independent, practical experience shows that they are actually strongly interrelated. This observation is a key point for this tutorial: its focus is on an integrated presentation of product line scoping and modeling technologies. The presentation illustrates this with the PuLSE³ method, which has been successfully applied in industrial practice.

Intended Audience: The tutorial aims both at researchers wishing to get a fresh view of the topic of product line modeling and scoping, and at practitioners wishing to gain an understanding of how they can integrate product line scoping and modeling techniques in their existing practices.

Tutorial 5: Component-based Product-line Engineering with the UML⁴
Presenter: Dirk Muthig, Fraunhofer IESE

Tuesday, August 20
Time: 1:30 p.m. – 5:00 p.m.
Room: Coronado A

Abstract: To date the component paradigm has only really penetrated the implementation and deployment phases of the software life cycle; it does not yet play a major role in the earlier analysis and design activities of large software projects. This tutorial will present a new method for component-based software engineering, known as KobrA, which supports a higher-level, model driven representation of components and thus enables the benefits of components to be realized throughout the software life cycle. Other distinctive features of the approach include the use of a product-line approach to develop and deploy component frameworks, a systematic, architecture-centric process that highlights the component composition hierarchy, and fully integrated quality assurance in the form of inspections and
Tutorial 6: Global Analysis: Developing Software Architecture Design Strategies to Respond to Variation and Anticipated Change

**Presenters:** Robert L. Nord and Daniel J. Paulish, Siemens

**Tuesday, August 20**

**Time:** 8:30 a.m. – 12:00 p.m.

**Room:** Coronado B

**Abstract:** The purpose of global analysis is to analyze the factors that influence the architecture and to develop strategies for accommodating these factors in the architecture design. Some of these influencing factors—which include market requirements, technologies, hardware, and business factors—affect the entire system, and some directly contradict other factors. In order to avoid major potential rework, they must be addressed from the beginning of high-level design. By applying the approach described in this tutorial, we believe that projects will have better understanding and control of the risks associated with alternative software architecture designs. Such an approach can be used to support the design of a single evolving system or variation within a product line.

**Intended Audience:** This tutorial is intended for experienced software engineers, architects, and project and technical managers.

Tutorial 7: Architecture-Centric Software Engineering

**Presenter:** Jan Bosch, University of Groningen

**Tuesday, August 20**

**Time:** 1:30 p.m. – 5:00 p.m.

**Room:** Coronado B

**Abstract:** This tutorial provides both an overview and an in-depth treatment of the issues surrounding architecture-centric engineering of software. Topics include software architecture design in the presence of existing components and infrastructure (top-down versus bottom-up), architecture evaluation and assessment, software artifact variability management, software product lines, and the role of the software architect. In addition to the technical perspective, the topics are discussed from process and organizational viewpoints, and are extensively illustrated by examples and experiences from many industrial cases.

**Intended Audience:** The expected audience is (1) software engineers and technical managers
considering the introduction of architecture-centric software development and evolution, and (2) researchers interested in the experiences collected by the tutorial presenter and his research group and the reflections made based on the experiences. Participants are assumed to have some experience with industrial software development.

1 PI is a trademark of Prosperity Heights Software.
2 Capability Maturity Model is registered in the U.S. Patent and Trademark Office.
3 PuLSE is a registered trademark of the IESE.
4 UML is a trademark of the Object Management Group.

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

Copyright ©2002 by Carnegie Mellon University
The Second Software Product Line Conference

Conference Workshops

Important: Workshop Attendance Criteria

Monday, August 19

- Workshop 1: Techniques for Exploiting Commonality Through Variability Management
- Workshop 2: Model Driven Architecture and Product Line Engineering
- Workshop 3: Mining and Reengineering of Legacy Assets for use in a Product Line
- Workshop 4: Future Visions of Interoperable Ground System Product Lines

Tuesday, August 20

- Workshop 5: The Fifth Department of Defense Product Line Practice Workshop
- Workshop 6: Managing the Architectural Evolution of Software Product Lines
- Workshop 7: Educator's Workshop

Workshop 1: Techniques for Exploiting Commonality Through Variability Management

Monday, August 19
Room: La Jolla
Time: 8:30 a.m. – 5:00p.m.

Variability is a quality factor expressing the ease with which existing software may be adapted and reused. This workshop will explore promising techniques for managing variability through a case study approach. The workshop will present attendees with a specific product line problem to solve. The goal of the workshop is to establish approaches for managing variability as a means of exploiting commonality across a product line. The problem statement will go out in advance of the workshop. We will take position papers in return and organize into subgroups:
● specific domains within the product line - UI, data base, embedded sensors, etc.
● product line practice areas - requirements management, architecture, tool support, etc.
● methods for handling variability - generators, aspect-orientation, UML extensions, etc.

Workshop Organizers:

Sholom Cohen
Software Engineering Institute
Pittsburgh, PA 15217
Phone: +1 412-268-5872
FAX: +1 412-268-5758
email: sgc@sei.cmu.edu

Bedir Tekinerdogan
University of Twente
The Netherlands
email: bedir@cs.utwente.nl

Mehmet Aksit
University of Twente
The Netherlands
email: aksit@cs.utwente.nl

Krzysztof Czarnecki
DaimlerChrysler AG
Research and Technology
Software Engineering Lab
Software Architecture Group (FT3/SA)
Wilhelm-Runge-Str. 11
89081 Ulm, Germany
email: krzysztof.czarnecki@web.de

Workshop 2: Model Driven Architecture and Product Line Engineering

Monday, August 19
Room: Carmel
Time: 8:30 a.m. – 5:00p.m.

Model Driven Architecture (MDA)[www.omg.org/mda/presentations.htm], the new OMG initiative, defines an approach to IT system specification. The approach separates the specification of system functionality from the specification of the implementation of that functionality on a specific technology platform. MDA addresses the complete life cycle of specifying, designing, deploying, integrating and managing IT applications placing models at the center of the development. The goal of this workshop is to analyze Model Driven Architecture as a technological proposition underlying product line engineering. The analysis will bring answers to the following questions:

● Why would MDA be a good technological approach for product line engineering?
● What are the characteristics of MDA that are of interest to a product line approach?
● How does MDA address product line issues such as: variability modeling, traceability, derivation supports, platforms configurations, etc.
● What is missing in MDA in order to address all the technical issues behind product line engineering?
Workshop Organizers:

Jean Jourdan (General Chair)  
Thales Research and Technology  
Domaine de Corbeville  
91404 ORSAY France  
Phone: +33 (1) 69 33 09 18  
email: jean.jourdan@thalesgroup.com

Serge Salicki  
Thales Research and Technology  
Domaine de Corbeville  
91404 ORSAY France  
Phone: +33 (1) 69 33 09 14  
email: serge.salicki@thalesgroup.com

Sergio Bandinelli  
European Software Institute  
Parque Tecnologico de Zamudio # 204  
E-48170 Bilbao Spain  
Phone: +34 9 4 420 95 19  
email: sergio@esi.es

Philippe Desfray  
SOFTEAM  
144 Avenue des Champs Elyssée  
75008, Paris, France  
email: Philippe.desfray@softeam.com

Desmond D'Souza  
Kinetium  
email: desmond@kinetium.com

Jean Bézivin  
Université de Nantes,  
email: Francejean.bezivin@sciences.univ-nantes.fr

Workshop 3: Mining and Reengineering of Legacy Assets for use in a Product Line

Monday, August 19  
Room: Del Mar  
Time: 8:30 a.m. – 5:00p.m.

Few product line efforts start from "green fields" because organizations have a substantial legacy base of existing software assets. But a lot of work must be done before an organization can reuse these legacy assets in a product line. The theme of this workshop is to examine what needs to be done.

The workshop's goal is to explore the current state of the practice in methods, tools and technology that can assist in carrying out the required work. The workshop will examine the following aspects of the problem of reusing legacy components:

- Understanding the legacy base through architecture reconstruction
- Decision making to identify what components can be reused
- Refactoring of the legacy software to produce reusable components
- Tool support for these activities

For more information please contact the workshop organizers:
Workshop Organizers:

Liam O’Brien
Software Engineering Institute
4500 Fifth Ave,
Pittsburgh, PA, 15213
Phone: +1 412-268-7727
FAX: +1 412-268-5758
email: lob@sei.cmu.edu

Dennis Smith
Software Engineering Institute
4500 Fifth Ave,
Pittsburgh, PA, 15213
Phone: +1 412-268-6850
FAX: +1 412-268-5758
email: dbs@sei.cmu.edu

Workshop 4: Future Visions of Interoperable Ground System Product Lines

Monday, August 19
Room: 7th Floor Boardroom
Time: 8:30 a.m. – 5:00p.m.

Many organizations have presented success stories and lessons learned regarding their ground system product lines; however, each organization presents results with respect to its own product lines. There has been little discussion, and no conclusions, regarding the possibility of interoperable product lines or product line components. The workshop will explore the current state of ground system product lines and begin to converge on a consensus regarding a future vision for ground system product line architectures and standards. The workshop will focus on the unique challenges and potential benefits of increasing interoperability among ground system product lines.

The workshop will focus on the issues above and include discussions of

- perspectives on the current state of ground system product lines
- lessons learned
- potential for and benefits of interoperable ground system product lines
- future visions for a component industry to support ground system product lines
- areas for potential standardization to support these visions

The workshop at SPLC2 will build upon and contribute to ground system product line work at the Ground System Architectures Workshops (GSAW). For more information on the SPLC2 workshop, please see http://www.aero.org/conferences/splc/ or contact the organizers.

Workshop Organizers:
Workshop 5: The Fifth Department of Defense Product Line Practice Workshop

Tuesday, August 20
Room: La Jolla
Time: 8:30 a.m. – 5:00 p.m.

This workshop will bring together participants from the DoD, other government organizations, and commercial practitioners who support these agencies. The SEI has held four previous DoD Product Line Practice workshops as separate events. This year we have decided to hold the DoD Workshop in conjunction with SPLC2 to both broaden and deepen the opportunity for software product line exposure for the DoD community. As in the past four DoD workshops, participants will share their product line development and acquisition practices in support of government product line efforts. Working groups will be established to make progress in specific areas. These focus areas will be finalized based on the participants' experiences and interests. A secondary agenda item is to obtain input as well as feedback and suggestions for the SEI's Software Product Line Acquisition - A Companion to the Framework for Software Product Line Practice.

If you have questions, please feel free to contact the workshop organizers:

**Workshop Organizers:**

Lawrence G. Jones  
Software Engineering Institute  
1155 Kelly Johnson Blvd., Suite 111  
Colorado Springs, CO 80920  
Phone: +1 719-548-4744  
FAX : +1 719-590-7652  
email: lgj@sei.cmu.edu

John K. Bergey  
Software Engineering Institute  
Phone: +1 215-348-0530  
email: jkb@sei.cmu.edu

Workshop 6: Managing the Architectural Evolution of Software Product Lines

Tuesday, August 20
This workshop brings together researchers and practitioners who wish to debate and extend their ideas in the area of managing the extension and re-design of the architectures of software product families that they develop. This workshop builds on the experience of the organizers in conducting similar events in this area and on the results reached during previous events:

ECOOP 2001 Workshop on Object-Oriented Architectural Evolution

WICSA 2001 Workshop on Architectural Viewpoints

To optimize the exposition of participants' views and to maximize interaction, the workshop adopts a question-and-answer style, whereby a number of questions concerning the architectural evolution of product families are proposed. We invite participants to submit well-reasoned answers to at least two of the questions. For details on the questions and the desired format of the answers please contact the workshop organizers.

Workshop Organizers:
Alessandro Maccari  
Nokia Research Center  
email: alessandro.maccari@nokia.com

Galal H. Galal  
University of North London  
email: galal@acm.org

Sergio Bandinelli  
ESI  
email: sergio@esi.es

Workshop 7: Educator’s Workshop

Tuesday, August 20  
Room: Del Mar  
Time: 8:30 a.m. – 5:00 p.m.

Research in software product line practice and the use of those ideas in industry have advanced to a point where product line topics are being introduced into a number of university curricula. The spreading use of software product line practices in industry is also resulting in an increase in industrial training courses. The purpose of this workshop is to provide a forum in which both academic and industrial educators can exchange experiences and ideas. The topics to be covered include, but are not limited to:

- The appropriate places in the graduate and undergraduate curriculum for product line topics,
- What specific topics or hands-on projects would be appropriate,
- What resources, case studies, artifacts, or tools are available to support instruction,
- Instructional techniques that have succeeded and those that have failed,
SPLC2 Workshops

- What academic prerequisites or experience should be required,
- What the appropriate role of product-line engineering is in the teaching of software engineering as a whole.

Participants from industry are strongly encouraged to apply.

For more information, please see the workshop web site: http://www.cs.clemson.edu/~johnmc/splc2.htm.

Workshop Organizers:

John D. McGregor  
Dept of Computer Science  
Clemson University  
Clemson, SC 29634  
Phone: +1 864-656-5859  
Fax: +1 864-656-0145  
email: johnmc@cs.clemson.edu  

Stuart Faulk  
Computer and Information Science  
University of Oregon  
Eugene, OR 97403-1202  
Phone: +1 541-346-1350  
Fax: +1 541-346-5373  
email: faulk@cs.uoregon.edu

Workshop Attendance

Workshops are by invitation only. Each workshop may have its own invitation criteria. The most likely criteria for attendance will be the submission and acceptance of a workshop paper. Workshop papers tend to be much less formal and comprehensive than main-conference papers.

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

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

Copyright ©2002 by Carnegie Mellon University
Tool Support for Product Lines: What We Have and What We Need

Moderator: David Weiss, Director of Software Technology Research, Avaya Laboratories

Panelists:
Charles Krueger, BigLever Software, Inc.
Bedir Tekinerdogan, University of Twente
Grady Campbell, Prosperity Heights Software
Daniel Simon, Universität Stuttgart, Institut für Informatik

We in the product line community have long decried the lack of tool support specifically geared to help manage the multi-dimensional production capability inherent in the software product line approach. After several years, though, some tool vendors are responding to the challenge. What tools are out there, and how do they compare with the needs (real or perceived) of product line practitioners?

This panel will explore that question. Each panel member has been chosen because of his experience in
building product-line-related tool technology. After a short opening statement by each, the session will turn to the audience for questions and discussion about where we should go from here.

---

**Software Product Lines: Crossing the Chasm**

**Moderator:** Stuart Faulk, Department of Computer and Information Science, University of Oregon

**Panelists:**
Dan Paulish, Siemens  
Scott Preece, Motorola  
Sergio Bandinelli, European Software Institute  
Linda Northrop, Software Engineering Institute

We know that software product lines represent a way to achieve true order-of-magnitude improvements in cost, schedule, and quality, something of a holy grail of the software engineering community. Given that, why isn't everyone using the approach where it is applicable?

The theme of this panel is how we as a community can help bridge the gap between those of us who know the benefits of software product lines, and industry and practice at large where the idea is still considered risky, unproven, and avant garde—where it is even considered at all.

The panel members each bring a unique perspective to the problem of transitioning software product line practice to the community. Technology transition organizations and academic institutions will represent the "push" side of the problem, while industrial practitioners will speak to the "pull" side of the equation.

Each panel member will make a short position statement, but the bulk of the session will be turned over to audience questioning and discussion.

---

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

Copyright ©2002 by Carnegie Mellon University
Anders Heie
Nokia Mobile Phones

Title: Global Software Product Lines and Infinite Diversity

Nokia Mobile Phones creates a wide range of products each year. Each of these has a unique combination of system software, style, and features. Not only is it an immense challenge to develop robust software to support all these products, but we also need to release them in time to market. Requirements pour in from across the world, as almost every operator is now starting to distance themselves from the mainstream. As if that wasn't enough, new features are constantly being thought up in-house as well. The capacity of our software and hardware is being constantly challenged, and it is essential to stay ahead of the game. In this talk, I will provide examples from Nokia illustrating how we handle such a variance of input and how we have attempted to create a software architecture that can support such demands. I will also touch on some of the processes that are necessary to get it all in place.

Anders Heie is a Software Specialist at Nokia Mobile Phones. He began at Nokia in 1995 working on the GSM protocol, first as a Configuration Manager and then moved on to the Software Architecture group. Here he helped develop parts of the core software that is now running throughout Nokia products. He regularly provides software training in Nokia Centers across the world.

Anders currently has 22 patents pending for Nokia in the US, three worldwide, and more in the works. He is a graduate of the Engineering Academy of Denmark.
The SPLC2 will host four demonstrations. The demonstrations will be held Wednesday, August 21st, from 5:30 p.m. to 6:30 p.m. Each demonstration will be approximately 1/2 hour in length.

**Rumi: A Tool Environment for Managing Product Alternatives**

Bedir Tekinerdogn,  
University of Twente

**Time:** 5:30 p.m. – 6:00 p.m.  
**Room:** Point Loma B

This demonstration presents the Rumi environment, which includes a set of tools for supporting the techniques of design space models (DSMs). DSMs provide a complementary technique to existing domain engineering and product line scoping techniques, to explicitly model and reason about domain alternatives. Design space modeling consists of a design space representation and semantic information for configuring and depicting the constraints on the alternatives.
**MetaSyn (TM): A Family of Tools for Building Product Families**

*Grady Campbell,*
Prosperity Heights Software

**Time:** 5:30 p.m. – 6:00 p.m.  
**Room:** Carmel

This demonstration exhibits a progression of prototype tool parts and assemblies that support the construction and instantiation of software product families. The focus of this demonstration will be on notations and mechanisms for the direct and explicit representation of variability as an orthogonal attribute characterizing a product family. A secondary emphasis will be on the resulting ease of deriving varied instances from a properly defined family using MetaSyn™ capabilities.

---

**GEARS (TM): Commercial Engineering Infrastructure, Development Environment, and Product Generator Enabling Rapid Adoption of Software Mass Customization**

*Charles Krueger,*
BigLever Software, Inc.

**Time:** 6:00 p.m. – 6:30 p.m.  
**Room:** Point Loma B

This demonstration illustrates the capabilities of BigLever Software GEARS™, a commercially available technology that supports software product line engineering through “software mass customization.” The demo will benefit practitioners and managers with responsibility for establishing and maintaining software product lines as well as researchers interested in learning more about the practical issues of engineering software product lines.

---

**Feature Analysis for Evolutionary Introduction of Software Product Lines**

*Daniel Simon* and Thomas Eisenbarth,  
University of Stuttgart

**Time:** 6:00 p.m. – 6:30 p.m.  
**Room:** Carmel

This demonstration shows an analysis of the graph-drawing tool, XFIG, using our feature analysis tools. Our tools analyze the monolithic legacy source code for product line initiation. The demonstration addresses mining and reengineering for product lines. Our tools guide developers through the legacy code in a feature-driven way. The results are produced by using the mathematically founded methods of “concept analysis.”

---

**Demonstration Discussions**

**Time:** 5:30 p.m. – 6:30 p.m.  
**Room:** La Jolla

---

MetaSyn™ is a trademark of Prosperity Heights Software.  
GEARS™ is a trademark of BigLever Software, Inc.
The SEI is a federally funded research and development center sponsored by the U.S. Department of Defense and operated by Carnegie Mellon University.

Copyright ©2002 by Carnegie Mellon University
Conference Announcement

This material is being posted by Carnegie Mellon University's Software Engineering Institute (SEI) on this Web site as a community service.

The Second Software Product Line Conference

August 19–22, 2002
San Diego, California

Call for Participation

The Software Engineering Institute is proud to sponsor the second Software Product Line Conference (SPLC2). SPLC2 is a forum for researchers and practitioners working in software product lines. Multiple successful workshops and SPLC1 have shown that the community is growing and its focus sharpening. SPLC2 is sure to enhance this maturation. The conference will feature technical papers, topical panels, tutorials, workshops, demonstrations, and birds-of-a-feather opportunities.

We look forward to interacting with you at SPLC2.

Linda M. Northrop
Software Engineering Institute
SPLC2 Conference Chair

Len Bass
Software Engineering Institute
SPLC2 Program Co-Chair

Henk Obbink
Philips
SPLC2 Program Co-Chair

Program Committee

Felix Bachmann
Software Engineering Institute

Stuart Faulk
University of Oregon

Frank van der Linden
Philips Medical Systems

Sergio Bandinelli
European Software Institute

Cristina Gacek
University of Newcastle

Nenad Medvidovic
University of Southern California

Don Batory
University of Texas at Austin

André van der Hoek
University of California, Irvine

Michael Moore
NASA/Goddard Space Flight Center

Joseph H. Bauman
Hewlett Packard

Jean Jourdan
Thales

Robert L. Nord
Siemens Research, Inc.

Günter W. Böckle
Siemens AG

Peter Knauber
Fraunhofer IESE

Scott Preece
Motorola

Technical Papers

Papers on all aspects of software product lines describing research results, research in progress, case studies, and industrial experience are being solicited for the conference. In particular, submissions that cover research or experience in the development and fielding of software product lines for complex systems, and that expose problems in the design, development, or evolution of software product lines are highly encouraged. We are primarily interested in papers that emphasize those aspects of product lines that differ from single system development.

Specific topics of interest include:

Understanding and managing variability in product lines

- product line scoping
- tool support
- requirements
- software architecture
- binding time
- lightweight technologies for managing multiple systems simultaneously

Business issues for product lines

- data about adoption costs, error costs, sustaining costs
- economic models
- lowering the up-front investment
- metrics

Organizational issues for product lines

- organizational structures
- personnel issues
- role of management
- institutionalizing a product line approach
- risk management

Product line life-cycle issues

- adoption models
Conference Announcement

- technology refreshment and insertion
- recognizing degradation in product lines
- mining and reengineering

Paper Submission Guidelines and Templates

Technical papers may be submitted as research papers (describing work in progress) or experience reports (summarizing the results of previous work). Each must describe original work that has not been published and must not exceed 7500 words.

Selected research papers and experience reports will be published by Springer-Verlag in conference proceedings. Papers will be accepted in MS Word format. Submissions should be in PDF or PostScript based on the following LNCS templates: sv-lncs.dot (for use with PC systems) or sv-lncs (for use with Macintosh systems). Please read the "Instructions for Using Author Template" sv-lncs.doc carefully. Also provided is typeinst.doc, the "Authors' Instructions" as an example input. All four files can be downloaded as a ZIP archive.

Technical papers will be accepted starting November 1, 2001.

NOTICE: The electronic submission service may be briefly unavailable on Monday mornings (during the hours of 1:00am EST to 8:00am EST) for system maintenance.

Panels

Paul Clements
Software Engineering Institute
Panels Chair

The program will include one or more panel sessions in which participants share their views on a specific topic or debate both sides of a stated resolution. The goal of a panel is to stimulate thought on related topics and encourage lively and engaging discussion among conference participants. Special consideration will be given to proposals that are creative or innovative in topic or presentation approach. Panel proposals must not exceed 500 words.

Each should contain

- a precise statement of the topic to be discussed or debated
- the names and affiliations of up to seven panelists (for whose participation you are willing to be responsible)
- a short statement about each panelist's qualifications to discuss/debate the proposed topic
- the steps the panel will take to actively engage the audience

E-mail panel proposals in plain ASCII format to clements@sei.cmu.edu by December 15th 2001.

Tutorials
Conference Announcement

Patrick Donohoe  
*Software Engineering Institute*  
*Tutorials Chair*

Tutorials provide a valuable opportunity for conference participants to expand their product-line knowledge and skills. Tutorials may focus on introductory product-line topics, such as how to introduce a product-line approach into an organization, or on more advanced applied topics such as industrial product-line engineering practices.

Tutorials will be held on Monday and Tuesday of the conference week, and will be presented in half-day or full-day sessions.

A tutorial proposal consists of two to three pages describing the topic, the plan for conducting the tutorial, and the presenters' backgrounds.

- The Topic section should include the title, the goals, and the intended audience for the tutorial. The topic should be described in detail, stressing its importance and timeliness.
- The Plan section should include the duration of the proposed tutorial (half or full day), a preliminary schedule of events including estimated times, a breakdown of the subtopics within the general topic (i.e., a detailed description of what the tutorial will cover), a justification of the tutorial for a product-line audience, and an explanation of how the tutorial will be conducted, including sample materials to be included in the tutorial notes.
- The Presenters' Backgrounds section should include relevant biographical information, and summaries of the presenters' technical, presentation, and tutorial experience.

E-mail tutorial proposals to **pd@sei.cmu.edu** by December 15th, 2001.

---

**Workshops**

Sholom Cohen  
*Software Engineering Institute*  

Kyo C. Kang  
*Pohang University of Science and Technology, Korea*  

**Workshop Co-Chairs**

Workshops provide a valuable opportunity for small groups of people with diverse perspectives to discuss topics of common interest. Interaction among participants is important, so participants must have informed positions based on experience.

Workshops can focus on research or applied topics. Innovative, controversial, or highly practical topics are particularly suitable for workshops. Workshop proposals on all aspects of product lines are encouraged. Topics could range from product line development methods such as domain analysis, product line requirements analysis, product line architectures to domain specific issues in areas such as telecommunications, embedded systems, etc. Each workshop will result in an SPLC workshop report that will provide an organized way of viewing the topic and will suggest directions for promising future research.
The workshops will be held on Monday and Tuesday of the conference week and will last one day. Most workshops should have 20 or fewer participants. Each workshop organizer will summarize the workshop discussions as part of a plenary panel on Thursday, August 22nd, 2002.

Workshop proposals should be 2-3 pages and should contain

- workshop name
- organizer names and affiliations
- an outline of the theme and goals of the workshop and its relevance to the conference
- a description of the desired number of participants, the participant solicitation and selection process, and any pre-workshop activities
- planned workshop activities, including details such as length of time, required rooms and facilities, and expected workshop outputs
- a brief description of each organizer's background, including relevant past experience with product lines and workshops

E-mail workshop proposals to sgc@sei.cmu.edu by December 15th, 2001.

Demonstrations

Felix Bachmann  
Robert Bosch GmbH  
Demonstrations Chair

Demonstrations provide an opportunity to describe work-in-progress, to display the results of software product lines, and to share the unique and interesting technical aspects of product line tools or systems. The demonstrators will be technical people who will seek active participation of and interaction with the attendees. The demonstrators will supply any equipment needed for the demonstration. Each demonstration will be given in a 30-minute session.

Demonstration proposals should be 2-3 pages in length and should include

- demonstrator name and affiliation
- demonstration name and description, including the relevance to software product lines and the experience to date
- a brief description of each demonstrator's background

E-mail demonstration proposals to fb@sei.cmu.edu by December 15th, 2001.

Electronic Submissions

NOTICE: The electronic submission service may be briefly unavailable on Monday mornings (during the hours of 1:00am EST to 8:00am EST) for system maintenance.

All papers must be submitted in MS Word using the Springer supplied templates. Submissions should be in PDF or
Conference Announcement

PostScript based on the following LNCS templates: sv-lncs.dot (for use with PC systems) or sv-lncs (for use with Macintosh systems). Please read the "Instructions for Using Author Template" sv-lncs.doc carefully. Also provided is typeinst.doc, the "Authors' Instructions" as an example input. All four files can be downloaded as a ZIP archive. Papers will be accepted starting November 1, 2001.

Send
Technical paper questions: ljb@sei.cmu.edu
Panel proposals to: elements@sei.cmu.edu
Tutorial proposals to: pd@sei.cmu.edu
Workshop proposals to: sgc@sei.cmu.edu
Demonstration proposals to: fb@sei.cmu.edu

For general information, contact Linda Northrop at lmn@sei.cmu.edu
For SPLC2 Web page information, contact Bob Krut at rk@sei.cmu.edu

---

San Diego

San Diego has nearly perfect year-round climate and terrain, from sunny beaches and lush foliage to rugged mountains and breathtaking desert. It is both a modern metropolis, with the accompanying cultural advantages, as well as a year-round resort. Home to many world class chefs, you will enjoy San Diego's dining and nightlife, shopping and entertainment, and cultural and leisure activities such as the city's 92 golf courses!

For more information about San Diego visit http://www.sandiego.org
The Second Software Product Line Conference

Registration Information

Register Now

Early Bird registration ends July 31, 2002.

Price Description:
- General Attendee
- Student Attendee
- Workshop Attendee

Important Date: July 31, 2002

Fees and Options
Refunds and Submissions
Hotel Information
Travel Information
Dietary Restrictions
San Diego

Important Date: July 31, 2002

- Last day to register at the Early Bird Rate
- Last day to register with a Purchase Order
- Last day to cancel conference registration minus a $50 Administrative Fee
- Last day for accepting dietary restrictions

Price Description
## General Attendee

<table>
<thead>
<tr>
<th></th>
<th>Early Bird Price (until July 31)</th>
<th>Just In Time Price (after July 31)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conference Only</td>
<td>$400.00</td>
<td>$450.00</td>
</tr>
<tr>
<td>Conference and Tutorial (1 Day)</td>
<td>$775.00</td>
<td>$875.00</td>
</tr>
<tr>
<td>Conference and Tutorial (2 Day)</td>
<td>$1,150.00</td>
<td>$1,300.00</td>
</tr>
<tr>
<td>Tutorial Only (1 Day)</td>
<td>$375.00</td>
<td>$425.00</td>
</tr>
<tr>
<td>Tutorial Only (2 Day)</td>
<td>$750.00</td>
<td>$850.00</td>
</tr>
</tbody>
</table>

## Student Attendee

<table>
<thead>
<tr>
<th></th>
<th>Early Bird Price (until July 31)</th>
<th>Just In Time Price (after July 31)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conference Only</td>
<td>$250.00</td>
<td>$300.00</td>
</tr>
<tr>
<td>Conference and Tutorial (1 Day)</td>
<td>$625.00</td>
<td>$725.00</td>
</tr>
<tr>
<td>Conference and Tutorial (2 Day)</td>
<td>$1,000.00</td>
<td>$1,150.00</td>
</tr>
<tr>
<td>Tutorial Only (1 Day)</td>
<td>$375.00</td>
<td>$425.00</td>
</tr>
<tr>
<td>Tutorial Only (2 Day)</td>
<td>$750.00</td>
<td>$850.00</td>
</tr>
</tbody>
</table>

## Workshop Attendee

<table>
<thead>
<tr>
<th></th>
<th>Early Bird Price (until July 31)</th>
<th>Just In Time Price (after July 31)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conference, Tutorial (1 Day), and Workshop (1 Day)</td>
<td>$925.00</td>
<td>$1,075.00</td>
</tr>
<tr>
<td>Conference and Workshop (1 Day)</td>
<td>$550.00</td>
<td>$650.00</td>
</tr>
<tr>
<td>Conference and Workshop (2 Day)</td>
<td>$700.00</td>
<td>$850.00</td>
</tr>
<tr>
<td>Tutorial (1 Day) and Workshop (1 Day)</td>
<td>$525.00</td>
<td>$625.00</td>
</tr>
<tr>
<td>Workshop Only (1 Day)</td>
<td>$150.00</td>
<td>$200.00</td>
</tr>
<tr>
<td>Workshop Only (2 Day)</td>
<td>$300.00</td>
<td>$400.00</td>
</tr>
</tbody>
</table>

## Dietary Restrictions

We will order special meals (lunches only) for attendees who indicate a dietary restriction (vegetarian, low-fat, kosher, diabetic, or shellfish allergy) on registration forms received by July 31, 2002. The event organizers must order special meals by the date, and therefore will be unable to accommodate special meal requests received after that date.
Registration Information

Fees and Options

All registration fees include meals and conference materials.

Terms and Conditions

Payment

The Early Bird rate will be honored until 12 midnight U.S. Eastern Time on July 31, 2002. Payment in full must accompany registration. Method of payment may not be changed after submission of registration. Acceptable methods of payment include the following:

Credit Cards

- MasterCard, Visa, American Express, or Diners Club
- Credit cards are processed upon receipt

Company or Personal Checks

- Made payable to SEI/CMU
- Must indicate registrant's name

Complete Purchase Orders

- Accepted only until July 31, 2002
- Must indicate "Advanced Payment Required"
- Signed by the designated fiscal officer in your organization
- For government purchase orders, we recommend using DD Form 1556 as a pre-paid training form, with "Advanced Payment Required" in block #37 and fiscal officer's signature in block #29

Refunds and Substitutions

Registration fees are not transferable to other SEI events. The deadline for refunds is July 31, 2002. To request a refund or send substitute attendees, registrants must email notification to registration@sei.cmu.edu by the deadline. The SEI will issue refunds (less a $50 administrative fee) for requests received in writing or via email by the deadline. No refunds will be issued after the deadline. Refunds will not be given for sessions missed due to acts of God. Registrants who do not request a refund in writing by the deadline are responsible for payment whether they attend or not. If they do not attend, the conference
material will be mailed to them after the conference.

Contact Us About Registration

For more information about registration, email registration@sei.cmu.edu or call 412-268-7388.

Online Registration

DISCLAIMER OF WARRANTIES AND LIMITATION OF LIABILITY

THIS ONLINE REGISTRATION SITE IS PROVIDED BY THE SEI AND ITS SUBCONTRACTORS ON AN "AS IS" AND "AS AVAILABLE" BASIS. THE SEI AND ITS SUBCONTRACTORS MAKE NO REPRESENTATIONS OR WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO WARRANTY OF FITNESS FOR PARTICULAR PURPOSE OR MERCHANTABILITY, AS TO THE OPERATION OF THIS ONLINE REGISTRATION SITE OR THE INFORMATION, CONTENT, MATERIALS, OR PRODUCTS INCLUDED ON THIS ONLINE REGISTRATION SITE. THE SEI AND ITS SUBCONTRACTORS WILL NOT BE LIABLE FOR ANY DAMAGES OF ANY KIND ARISING FROM THE USE OF THIS ONLINE REGISTRATION SITE, INCLUDING, BUT NOT LIMITED TO DIRECT, INDIRECT, INCIDENTAL, PUNITIVE, AND CONSEQUENTIAL DAMAGES.

Register

Hotel

Hotel Reservations

Nestled at the water's edge on spectacular San Diego Bay, the Sheraton San Diego Hotel and Marina offers panoramic views of the bay and the downtown city skyline. Some of the country's most unique attractions are within a 10-minute drive of the hotel - including Sea World, the world famous San Diego Zoo, historic Old Town, Balboa Park and Seaport Village.

Sheraton San Diego Hotel and Marina
1380 Harbor Island Drive
San Diego, CA 92101
Phone: 619-291-2900
Fax: 619-692-2337
Registration Information

Web site: www.sheraton.com/sandiegomarina

Rates

<table>
<thead>
<tr>
<th>Room</th>
<th>Single Rate</th>
<th>Double Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>$199.00</td>
<td>$209.00</td>
</tr>
<tr>
<td>Premium</td>
<td>$219.00</td>
<td>$229.00</td>
</tr>
</tbody>
</table>

Government Rates

<table>
<thead>
<tr>
<th>Room</th>
<th>Single Rate</th>
<th>Double Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>$99.00</td>
<td>$99.00</td>
</tr>
</tbody>
</table>

Travel Information

The Sheraton San Diego Hotel and Marina is easily accessible from Lindbergh Field International Airport. For a map and directions, please visit the Sheraton San Diego Hotel and Marina Web site at: www.sheraton.com/sandiegomarina For general information, please call the San Diego Convention and Visitors Bureau at 619 / 232-3101 or see its Web site at www.sandiego.org.

San Diego, California

California's second largest city and the United States' seventh largest, San Diego boasts a citywide population of nearly 1.3 million residents and more than 2.8 million residents countywide. Within its borders of 4,200 sq. miles, San Diego County encompasses 18 incorporated cities and numerous other charming neighborhoods and communities, including downtown's historic Gaslamp Quarter, Little Italy, Coronado, La Jolla, Del Mar, Carlsbad, Escondido, La Mesa, Hillcrest, Barrio Logan, Chula Vista and more.

Known for it's near-idyllic climate, 70 miles of pristine beaches and dazzling array of world-class family attractions, including the World-Famous San Diego Zoo and Wild Animal Park, Sea World San Diego and LEGOLAND California, San Diego offers a wide variety of things to see and do, appealing to guests from around the world.

In San Diego's East County, the terrain varies from gentle foothills to mile-high mountains and the historic mining town, Julian, down to the 600,000-acre Anza Borrego Desert State Park, offering nature-conscious visitors endless opportunities to hike, camp, fish, observe wildlife and much more. In San Diego's North County, the land produces quantities of flowers as well as quality grapes that become...
excellent wines, which are served at some of the most elegant restaurants and resorts in the region. Along the west, 70 miles of Pacific Ocean coastline not only supports year-round outdoor recreation, such as surfing, boating, sailing and swimming, but also important scientific research at the Scripps Institution of Oceanography. To the south, it's a whole different country, Mexico, featuring its own cultural offerings in various towns along the border and coastline, including Tijuana, Rosarito and Ensenada.

San Diego's arts and culture and culinary arts are making a name for themselves, both nationally and internationally. Balboa Park, the largest urban cultural park in the U.S., features 15 museums, numerous art galleries, beautiful gardens, the Tony Award-winning The Globe Theatres and the World-Famous San Diego Zoo. The region is also a breeding ground for the hottest, new talents of culinary arts, who prepare award-winning meals in many of the region's 6,400 eating establishments.

San Diego County also features 92 golf courses and a variety of exciting participatory and spectator sports, beachfront resorts and luxury spas, gaming, a dynamic downtown district, annual special events and unique holiday offerings, multicultural festivals and celebrations, colorful neighborhoods and communities, a rich military history, accessibility for travelers with disabilities and much more.

The most difficult decision to make regarding a vacation to San Diego is determining what to do and see among the region's vast and diverse variety of offerings. San Diego County offers a vacation experience for everyone.

For more information about things to see and do in San Diego, call the San Diego Convention and Visitors Bureau at 619-232-3101 or see its Web site at www.sandiego.org.

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

Copyright ©2002 by Carnegie Mellon University
Biographical

- Anders Heie
- Software Specialist
- Nokia Copenhagen 3 years
- Nokia San Diego 4 years
- I’ve been working with...
  - UI Software Architecture
  - General SW Architecture
  - Messaging
  - Java
  - Provide input to SW Strategy
- 22 Pending Patents

Public
Group Project

- Define a product line with ~25-30 new products a year
- Across products, you must support:
  - Varying number of keys
  - Varying Display Size
  - Varying sets of features
  - A number of languages, and input methods
  - Backwards Compatibility to accessories
  - Different protocols and API’s
- And of course make them segmented (low end, high-end, …).
- Each Feature must be:
  - Configurable (On/Off, various settings)
  - Able to change behavior after product release
  - Plug’n’playable
Nokia Facts

- This year, we have released 12 new products (lots more on the way)
- We sell phones in more than 130 countries
- We support 58 Languages, amongst which are:
  - Latin Languages
  - Arabic
  - Chinese (With variants)
  - Thai
  - Hebrew
- We support multiple Protocols:
  - CDMA, TDMA, AMPS, GSM, GPRS, and more...
- The HW is constantly changing, and the SW is constantly expanding to provide features
- We have several different UI Series to support
The Language Challenge

- Language Input Methods
  - Latin (Letters, Left to Right)
  - Arabic (Right to Left, except when adding Latin)
  - Chinese (Entered by selecting the sounds that makes up the symbol through a series of strokes)
  - Thai (Entered similar to Chinese. But different logic)
  - Hebrew (…)
- T9 Libraries complicates this further
Instant Language Change

- If a user changes the active language, all texts must change instantly:
Language Challenge Abstract

• So what is the real challenge with languages?
  If each feature has to know about the language, the code becomes complicated and harder to maintain.
• The solution:
  • Separate the language knowledge from the Code.
• Abstractly speaking:
  • Separate Appearance and Behavior

This principle easy to formulate, but tough to achieve.
Observer pattern

**Abstraction** between **Appearance** and **Behavior**

```
Appearance  Appearance  Appearance

A = 50
B = 30
C = 20

Behavior
```

Public

http://www.sei.cmu.edu/SPLC2/keynote_slides/keynote_10.htm
Language Solution

- The solution:
  1. Create a text database. All texts are defined in there, and logical references are used.
  2. Texts are resolved during Display Updates.
  3. Split Input methods into Components.

- Results:
  - Whatever language is chosen, the Code doesn't change.
  - Adding a language is transparent
  - Adding an input method is transparent
UI Interface

- The User Interface is controlled through a Window Manager
- It is designed to separate Behavior and Appearance to the largest extent possible
- The Window Manager controls a set of Graphical components:
  - Components are Objects
  - Simple, small and well defined
- The Window Manager allows:
  - Clients can define abstract Input/output
  - UI behaviour that is irrelevant to Feature Behaviour can be modelled with Component interactions.
Appearance & Behavior

- Separating Appearance & Behaviour with Window Mgr.

Behavior

Client

Window Manager
(UI Controller)

Appearance

User Input/Output

Abstraction

... for (a = 0; a <= 10; a++)
    AddProfileMenu(MenuArray[a]);
...
The Hardware Challenge

- Each of our product families have unique hardware:
  - Keys
    - One, Two or Three Soft buttons (Creates radically different UI designs)
    - Full Keypads (Like Communicator for instance)
  - Sound Playback (Ringing 'beeps' versus MIDI)
  - Display size
  - Color depth (B&W, Grey Scale, Color)
  - Connections: Infrared, Bluetooth, …

- Plus, our HW must be backwards compatible with accessories already on the market.
Hardware Solution

- The HW problem is really another case of abstraction:
  - We need the code to be independent of the HW.

- Again, our Window Manager comes to the rescue:
  - We allow the abstraction of physical input/output to pure data.
Hardware Abstraction Example

- When waiting for user selection from a list, what does the client really want?

1. The keypress? "Keyboard: Left Softkey"
2. The list index? "list index: 5"
3. The information that the selection represents! "List Recipients"

- By supporting this abstraction, we allow the Window Manager to disconnect the HW (The Key) from the relevant data "List Recipients".
User Interface

- Separating Appearance & Behaviour with Window Mgr.

Behaviour

Client

Window Manager UI Controller

Client

Appearance

User Input/Output

Abstraction

case MenuNormal: SelectFeature(Normal);
case MenuSilent: SelectFeature(Silent);
case MenuOutdoor: SelectFeature(Outdoor);

Public

Nokia
The Feature Challenge

- Feature Variability
  - Protocols
  - Operators
  - Hardware
  - Product Style (Series XX)
- New features are created continuously
- Most features change over time
- Feature Memory usage and Speed are always important
Client/Server Architecture

• Enables a nice encapsulation of resources, and it allows adding and removing Entities (Features) easily
• Supports Service Discovery nicely
• Focuses on Interfaces.
• This is essentially an instance of the Observer Pattern. The Clients are the Observers, the Servers are the Subjects. This pattern decouples the actual Data Management from the presentation.
• The Client/Server model allows for Plug’N’Play entities which can be added to the system at runtime
Feature Selection

- With all our variants, it is essential to be able to select features. There are a couple of ways to handle that...
  1. Static Selections.
  2. Dynamic Selections
- Selecting a full feature can be done simply by including it in the build.
- Selecting just parts of it requires developer intervention (to manage the variability):
  - Add compile time flags,
  - React on dynamic settings,
  - Add configuration,
  - Provide for dynamic Service Discovery,
  - ...
Feature Selection Example

Product #1
Operator B
Phone-Book
SMS
Voice Calls
Browser

Product #2
Operator A
Clock
Color Display
Call Cost
Calculator
Voice Recognition

All Features

http://www.sei.cmu.edu/SPLC2/keynote_slides/keynote_23.htm
SW Lines

- Nokia Facts:
  - Global Company
  - Multiple R&D Centers spread across Countries & Continents
  - We have lots of Common SW

- We have SW Lines for various levels of SW:
  - DSP,
  - Architecture,
  - User Interface,
  - Various Feature Groupings,
  - etc.

- A SW Line is a group of people developing a specific set of features for a wide range of products. Ever so often a new release is created for others to use.
SW Line Considerations

- Each SW Line can potentially deliver to all of our products, so...
- Each deliverable must be tested in as many configurations as possible.
- A Feature Release that works for one product, can potentially crash another.
- A Global DB of Dependencies are maintained to ensure all affected parties are up-to-date on changes.
- Changes are documented and can be reviewed by the affected products to ensure interoperability.
Requirement Origins

New Technologies
Operator Requirements
Errors
Cultural Values
Nokia Development
Backward Compatibility
Protocols

Product A
Product B
...

Public

http://www.sei.cmu.edu/SPLC2/keynote_slides/keynote_27.htm
Errors

- Errors must be controlled tightly.
- Our Database of Dependencies ensures that all interested parties are duly informed
  - Errors Fixes can be distributed faster to projects.
- Errors are tested modularly at the appropriate level:
  - Example 1: A SW Line delivering common SW would test their own module before a release
  - Example 2: An error found in code originating from a SW Line, is sent back for correction.
Prioritizing

- While dependencies are known to a SW Line, priorities are not always as clear.
- A Strategic team determines a prioritized set of customer products.
- The priorities is not only dependent on products, but also on the impact a change has on existing Code and Schedules.
- Without this team, each line runs the risk of doing work that is not needed for a long time, while ignoring immediate needs.
Example: Operator Requirements

- Operator Specific features (Shortcuts, SMS formats, Proprietary Menus, …)
- New market driven features (MMS, GPS, …)
- Specialized Functionality of existing Features (SMS, Games, Browser, Data, …)

- Each of these requirements has the potential to ‘eat up’ a time schedule if introduced in the middle of product development
- Operator requirements typically starts out with local solutions.
Suggestion: Think Ahead

- Always 'think ahead'
- Brainstorm with colleagues about all the possible future uses of features
- Consider the impact of new technologies
- Consider all the variables that should be configurable
- Never hardcode anything
- While you don't have to implement all of the 'futures', take them into account in your design. It will save a lot of major design changes later
Suggestion: Requirements

- Requirements:
  - Must be clear. This sounds obvious, but it is really hard to do. The best requirements are not only clear, but also focusing on the general issues rather than the particulars. Consider these two:
    1. The Indicator area is 20 pixels high
    2. The Indicator area is as high as the largest icon + header & footer pixels.
  - Include Variability at the Requirements level
    - Specify commonality, and just as important;
    - Specify differences!
    - Some Variability Spheres: Products/Countries/Operators/Features/Protocols
Suggestion: Architecture

- Modular Architectural Design
  - Allows the separation of specialized and common components
- Scope Products according to Legacy
  - Take care not to scope new products that cannot easily be derived from existing products
  - When you do need to create radically new products, make sure the architectural impact is understood
- Control Change Requests
  - Change Requests affecting the Platform, should be approved by Architecture management
Managing the Challenges

1. Drive changes
   • Participate in technology groups
   • Invent new features

2. Anticipate changes:
   • New technology trends
   • New features hitting the market
   • Write flexible code, have flexible processes
   • Create and *Maintain* a good SW Architecture
   • On a developer level – Be creative, anticipate changes. Think Crazy - Act Sane

Public
Summary

- Problems
  - Sheer number of products and variants
  - Perpetual changes in requirements
  - Lots of Variability Points

- Solutions
  - Flexible Architecture
  - Separate Appearance and Behavior
  - Encourage Supportive Processes
  - Clear requirements (Common/Specialized)
  - Common Deliverables from Line organization
  - *Act like a Scout and Be Prepared*
Thank you for your attention.
What chasm?

- “Every software company does product line development” (Charlie Krueger, yesterday)
- My first software gig (1967) involved clone-and-own, asset mining, and parameterized generators
- So, the rest of this is talking about formalized SPL (planning, methods, and tools)
If we all do it, is SPL a secret?

- Agile methods took two years to common knowledge – no special tools or training and it panders to what engineers like to do
- O-O took 20 years to catch on, despite our love for anthropomorphism
- Modeling isn’t common, despite UML buzz
- Colleges don’t teach version control – we expect them to teach SPL?
Why might we cross the chasm?

- Would we be able to do something we can’t do, that produces more revenue than it costs?
- Would we save more, compared to existing methods, than additional costs?
- Is the pitch so compelling that we just really want to buy a bridge?
What’s on the other side?

- Cross-over chart isn’t very convincing – if you know enough to think about SPL, you aren’t building similar products from scratch
- Comparison has to be against what we actually do, not against start-from-scratch
- Tool cost is only a start – all the formal SPL methods involve a lot of additional models that need to be built and maintained
Who needs to cross the chasm?

• Need guidelines and economic models:
  – How much planning to do?
  – How many models to do?
  – What tools do I need?
  – What can I expect to get back?

You have to care about tenths of a knot to think a $100K sail is a necessity.
So, how do we cross the chasm?

• First, convince people there's a chasm
  – Measure it in widely-accepted assessments
  – Build CEO buzz
• Then convince them there's a bridge
  – More success stories – using packaged methodologies
  – Include it in university programs and business schools
• Then sell them a bridge
  – Major research funding to get to validated tools and models
  – Get Rational to sell it
• And people will cross it (or say they did)
Contents

• The Goal

• The Steps of Product Line Adoption

• Adoption Plan
  • Adoption Strategies
  • Product Line Maturity Assessment
  • Product Management Assessment

• Institutionalization
The Goal

• Precondition
  ▪ An organization has already products in its markets
  ▪ It wants to switch to a product line engineering approach
  ▪ It may integrate its previous products into the product line

• Goal
  ▪ Determine all activities for switching to a product line:
    ▪ Convincing all affected personnel
    ▪ The transition process
    ▪ Risks
    ▪ Strategies
    ▪ Organization
  ▪ Determine how to launch the product line
  ▪ Determine how to institutionalize the product line in the organization

• Postcondition: A successful product line engineering approach
The Steps of Product Line Adoption

1. Develop business cases
2. Get (go/no-go) decision
3. Develop adoption plan
4. Launch and institutionalize (iteratively)

Figure from "The Business Case Web Site" http://www.solutionmatrix.com/

Product Line Adoption
## Roles to Consider for Business Cases

<table>
<thead>
<tr>
<th>Role</th>
<th>Interest</th>
<th>Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manager</td>
<td>cost-based</td>
<td>budgeting, staffing</td>
</tr>
<tr>
<td>Product manager</td>
<td>revenue-based</td>
<td>marketing, customer analysis</td>
</tr>
<tr>
<td>Engineers</td>
<td>technology-based</td>
<td>developing assets</td>
</tr>
<tr>
<td>Project managers</td>
<td>milestone-based</td>
<td>delivering assets</td>
</tr>
<tr>
<td>Quality assurance</td>
<td>quality-based</td>
<td>product quality</td>
</tr>
</tbody>
</table>

*Different business cases for major roles!*
Adoption Plan

• Input:
  • Business goals
  • Information about current products, markets, customers, staff

• Contents:
  • Characterization of the organization's current state
    • process
    • staff + expertise
    • partners (e.g. subcontractors, universities, ...)
    • organization structure
    • project management methods
    • engineering methods
    • ...
  • Characterization of desired state
  • Strategies, objectives, activities to get from current to desired state
Adoption Strategies, Overview

- Incremental Introduction
- Pilot first
- Big bang
Adoption Strategies: Incremental Introduction

- Expanding scope:

- Incremental investment:

- Tactical
  - May start informally
  - Scoping and planning: formally
  - Results must be measurable and predictable
Adoption Strategies: Pilot First

- New product, tactical
- Prototype
  - both: either first member of a new product line
  - or extension of a series of existing products.
  - Apply appropriate methods of product line engineering
    - e.g. architecture recovery
- Toy product
  - reduced risk
  - close to real products
  - plan to re-use parts for later „real” products
Adoption Strategies: Big Bang

- Management must be convinced of PLE
- Achieving business results early is essential
- Most important activities:
  - Thorough scoping and planning
  - Measurements
  - Organization planning
  - Risk analysis
  - Milestones for reviews
Adoption Plan - Steps for all Strategies

- Introduce measures for
  - transition progress
  - comparing the development to previous product developments
- Plan monitoring and data collection
- Perform PL maturity assessment
- Milestones for review points
- Adoption review:
  - evaluate current state of adoption, using above measures
  - evaluate PLE approach compared to conventional engineering
  - adjust transition process accordingly
  - change adoption process and plan where necessary
  - update investment plan, organization plan, project management

Training plan
Product Line Maturity Assessment

• Similar to CMM, but result not a single number (CMM level), rather an evaluation of current state

• Consider:
  - Market characterization
  - Competition in the market
  - Degree of globalization
  - Management commitment
  - Expertise in product-family engineering (warm-start vs. cold-start)
  - Organizational maturity and stability
  - Staff turnover
  - Process maturity
  - Domain expertise and staff competence
  - Existing assets
  - Software infrastructure
  - Contractors' expertise and reliability
Results of a Product Management Assessment

Evaluation profile

Strengths / Weaknesses

Measures

Activity Portfolio

Product Line Adoption
Institutionalization

Product line approach works only if it lasts only

- stable and indispensable
- fixtures installed
- incentives
- funding structure
- measures installed and used
- policies, procedures, conventions among all stakeholders
- resources distribution
- Champions and angels

Exit Slide Show

Previous  Next

1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17
Institutionalization

Social arrangements
- persist
- hard to change
- people believe they will persist

Stable arrangements
- ease cooperation
- routine behaviors
- constrain different behaviors
- Change is localized
- people believe the PL will persist
- PL framework will guide cooperation
Governance Polarities of an Internal Product Line

- Summary
- Product Line History
- Polarities
- Polarities in your Product Line?
- Experience
- Challenges

9/10/2002

Software Product Line Conference 2002

Summary

• Polarity analysis is an organization change agent’s tool
  • toughest product line problems may not be technical, but organizational
  • polarity language is common, neutral terminology when aligning organizations

• Benefits of polarity analysis for product line governance planning and assessment:
  • acknowledges and addresses the tensions between primary business points of view
  • extends governance readily as product composition changes

9/10/2002  Software Product Line Conference 2002
Governance Polarities of Internal Product Lines

Business Products
Business Products: 10s

Development and Production Sites
Sites: 10s
Users: 1000s

Business Processes
Business Processes: 10s

Deployed Business Processes
Everett
Renton
Wichita

Delivered Business Products
737
787
777
Sonic Cruiser

Deployed IT Product Lines
Eng
Mfg
Material

IT GOVERNANCE SCOPE

Shared Architecture
Shared Products
Shared Processes

Aligned Single Core

Design
Structures
Leads

Software Product Line Conference 2002
9/10/2002
**Polarity**

- **Definition**: two principles which are both true, but which conflict

- **Polarity examples**:

  ![Diagram showing the polarity between Ruthless with time and Gracious with People, and Large enough to cover the subject and Short enough to be interesting.]

9/10/2002  Software Product Line Conference 2002
Holding Polarity

Approach

- Hold polarities; solve problems. Can’t solve polarities. Create Polarity Map to clarify poles to be held.

- Identify canary – first person to chirp, if overemphasizing opposing pole.

- Language: “A AND B”, not “A versus B”.

9/10/2002  Software Product Line Conference 2002
Polarity Description Tool: Polarity Map

<table>
<thead>
<tr>
<th>Pole A</th>
<th>Pole B</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>• Benefits of emphasizing Pole A</td>
<td>• Benefits of emphasizing Pole B</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>• Negative consequences of overemphasizing Pole A</td>
<td>• Negative consequences of overemphasizing Pole B</td>
</tr>
</tbody>
</table>

9/10/2002
Software Product Line Conference 2002

Previous Next
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

Exit Slide Show
Governance Polarities of Internal Product Lines

**Polarity Map:**  PART ← WHOLE

**Benefits:**
- Independence
- Freedom
- Individual Focus

**Benefits:**
- Integration
- Group Synergy
- Systemic Focus

**Negative Results of over emphasis:**
- Fragmentation
- Inequality
- Entrenchment

**Negative Results of over emphasis:**
- Lose Individual Creativity
- Lose Individual Initiative
- Excessive Conformity

**Examples:**
- NATO
- EC92
- Euro

**Example:**
- Feudal Europe
<table>
<thead>
<tr>
<th>Polarity Examples</th>
<th>PART ↔ WHOLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confederacy</td>
<td>States</td>
</tr>
<tr>
<td>Articles of Confederation: Poorly held polarity</td>
<td></td>
</tr>
<tr>
<td>Federal Government</td>
<td>State Government</td>
</tr>
<tr>
<td>U.S. Constitution: Well held polarity</td>
<td></td>
</tr>
</tbody>
</table>

9/10/2002
Software Product Line Conference 2002

Polarity Examples

+ effective business processes (design anywhere, build anywhere)
+ harvest benefits of initiatives

Technology

+ up-to-date tools
+ harness benefits of technology

Business

+ effective, agile governing bodies across globe

Organization

+ effective preparation for changes
+ clear roles and responsibilities

People

+ effective governance processes
+ coordinated, disciplined daily management

Process

Software Product Line Conference 2002

9/10/2002
Structure Analysis
for Airplanes

InfraStructure Analysis
for Organizations, e.g. IT Product Line

Carry people

Airplane Stress Analysis:
Ignore a stress point → risk failure

9/10/2002 Software Product Line Conference 2002

Infrastructure Stress (Polarity) Analysis:
Ignore a stress point → risk failure
Polarities of a Product Line

- Customer-Focus ↔ Product-Focus
- Customer-Focus ↔ Product-Focus ↔ Architect
- Effective Team ↔ Representative Team
- Product Line ↔ Product Family

LEGEND
Part ↔ Whole
Self ↔ Other

“the Software Problem”
NATO Software Conference, 1968
“software is late, costs too much and
does not do what we want”

No Silver Bullet
Fred Brooks, 1987

Corroboration:
software delivery management
is a polarity to be held,
not a problem to be solved with
a silver bullet
### Customer-Focus (CF) : Product-Focus (PF) Polarity Map

<table>
<thead>
<tr>
<th>Customer-Focus</th>
<th>Product-Focus</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>+</strong></td>
<td><strong>+</strong></td>
</tr>
<tr>
<td>• listen for gems</td>
<td>• reliable</td>
</tr>
<tr>
<td>• respond to gems</td>
<td>• predictable</td>
</tr>
<tr>
<td>• strong product support</td>
<td>• on-time</td>
</tr>
<tr>
<td>• product well-deployed</td>
<td>• on-$$</td>
</tr>
<tr>
<td><strong>-</strong></td>
<td><strong>-</strong></td>
</tr>
<tr>
<td>• product reqts change too rapidly</td>
<td>• product does not match customer need</td>
</tr>
<tr>
<td>• standard bearer: fail to harvest benefits</td>
<td>• technology push</td>
</tr>
<tr>
<td>• process transparent</td>
<td>• crusader: too much techno change</td>
</tr>
<tr>
<td>• too few products</td>
<td></td>
</tr>
</tbody>
</table>

#### CANARIES

<table>
<thead>
<tr>
<th>System Mgr</th>
<th>Process CF Mgr</th>
<th>Product CF Mgr</th>
<th>Program CF Mgr</th>
<th>Product Line CF Mgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Mgr</td>
<td>Process PF Mgr</td>
<td>Product PF Mgr</td>
<td>Program PF Mgr</td>
<td>Product Line PF Mgr</td>
</tr>
<tr>
<td>Project</td>
<td>Process</td>
<td>Product</td>
<td>Program</td>
<td>Product Line</td>
</tr>
</tbody>
</table>

9/10/2002
Software Product Line Conference 2002

---

10/17/2008 1:51:20 PM
Hold the CUSTOMER-FOCUS ◀-► PRODUCT-FOCUS Polarity

System Owner
- Provides $$
- Chairs St Comm

Steering Committee
- Managers of orgs where product deployed (customers)

Customer-Focus Manager
- Brokers deployment
- Brokers requirements

Product-Focus Manager
- Brokers commitment to build product
- Tracks schedules and resources

Manage customer-supplier tension: capability AND affordability

9/10/2002 Software Product Line Conference 2002
Project Polarities not held?

Impact:
System Manager only meets to Project Manager

Issue:
Architect role reports only to Project Manager

9/10/2002 Software Product Line Conference 2002
Holding Product Line Polarities

Customer-Focus Manager

Product-Focus Manager

Rationale:
Architecture deliverables are system-wide

9/10/2002
Software Product Line Conference 2002
Internal Product Line Polarities with Multiple Products

System Owner
- Provides $$
- Chairs Strng Comm

Steering Committee
- Managers of orgs where product deployed (customers)

Customer-Focus Manager
- Brokers deployment
- Brokers requirements

Product-Focus Manager
- Brokers commitment to build product
- Tracks schedules and resources

Architect
- Define and monitor shared architecture

9/10/2002

Software Product Line Conference 2002
Product Line Polarity Experience Summary

Customer-Focus <--- Product-Focus

- Saved Money
- Supported Customer Business Process

- Met product line schedules

9/10/2002
Software Product Line Conference 2002

###Shared Products

<table>
<thead>
<tr>
<th>Successful</th>
<th>Unsuccessful</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Consistent requirements statement from System Manager and Project Manager.</td>
<td>• Individual projects could not agree on shared development components.</td>
</tr>
<tr>
<td>• Factored out and managed shared, tightly coupled components.</td>
<td>• Integrated schedule management never quite implemented.</td>
</tr>
<tr>
<td>• Identified stewards for shared, loosely coupled components.</td>
<td></td>
</tr>
</tbody>
</table>

![Customer Focus ↔ Product Focus](image_url)

9/10/2002 Software Product Line Conference 2002
Shared Architecture

<table>
<thead>
<tr>
<th>Successful</th>
<th>Unsuccessful</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Repetitively assessed family member architectural progress.</td>
<td>• Not all products were able to identify architects.</td>
</tr>
<tr>
<td>• Mitigated risk of changes through product line architect reviews.</td>
<td>• Architect roles, responsibilities, and reporting relationship were ill-defined.</td>
</tr>
<tr>
<td>• Product line architects expended most energy on core components.</td>
<td></td>
</tr>
<tr>
<td>• Instituted a consistent forum for architecture issues.</td>
<td></td>
</tr>
</tbody>
</table>

Shared Processes

<table>
<thead>
<tr>
<th>Successful</th>
<th>Unsuccessful</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Assessed IT processes for usage and benefits.</td>
<td>• IT processes did not have clear stewards, making usage and benefit assessments uneven.</td>
</tr>
<tr>
<td>• All processes formally inspected and walked through.</td>
<td>• Processes not as clearly appreciated as technology.</td>
</tr>
<tr>
<td>• “at least they are talking”, not armed camps</td>
<td></td>
</tr>
</tbody>
</table>

9/10/2002  Software Product Line Conference 2002
Product Line Polarity Challenges

**Part ↔ Whole**

- Uneven process maturity across product development sites
  - reconcile at project level
  - common, neutral IT process architecture

- Scope constantly changing:
  - Extensibility is key architecture principle

- Largest Product Family dominates decisions
  - uniformly apply architecture monitoring

9/10/2002
Software Product Line Conference 2002
Product Line Challenge

* Quad-pole: four competing customer perspectives (sites, product family, business process, business program), each a PART ↔ WHOLE polarity, fragment the Product Line customer voice:
- currently, “people-strong; process-weak”, so attend SPLC2 for answer
Summary

- Polarity analysis is an organization change agent’s tool
  - polarity language is common, neutral terminology when aligning organizations

- Benefits of polarity analysis for product line governance planning and assessment:
  - acknowledges and addresses the tensions between primary business points of view, starter kit of four Product Line polarities
  - extends governance readily as product composition changes

Adopting and Institutionalizing a Product Line Culture

Günter W. Böckle, Siemens AG
Jesús Bermejo, Telvent
Peter Knauber, Fraunhofer IESE
Charles Krueger, BigLever Software, Inc.
Julio Cesar Leite, Pontifícia Universidade Católica do Rio de Janeiro
Frank van der Linden, Philips

Abstract: The strengths of product line engineering have been described before. But how can an organization make the move from developing one-off products to product line engineering without major interruptions in the day-to-day work? This paper describes how to perform the transition to product line engineering and lists the various strategies for such a transition. It describes how to create an adoption plan and how to institutionalize product line engineering in an organization.
Jan Bosch, University of Groningen

**Abstract:** Software product lines have received considerable adoption in the software industry and prove to be a very successful approach to intra-organizational software reuse. Existing literature, however, often presents only a single approach towards adopting and evolving a software product line. In this paper, we present an overview of different approaches to architecture-centric, intra-organizational reuse of software artifacts. We relate these to maturity levels for product line artefacts and organizational models.

---

**Adaptable Components for Software Product-Line Engineering**

John Brown, Queen's University of Belfast
Ivor Spence, Queen's University of Belfast
Peter Kilpatrick, Queen's University of Belfast
Danny Crookes, Queen's University of Belfast

**Abstract:** This paper explores techniques for implementing adaptable software components. Such techniques can greatly facilitate the implementation of software product lines. The techniques we present allow the construction of large, transparently adaptable components via composition and parameterisation. Functional and structural adaptation, to any level of nesting, is achieved at the point of instantiation via recursive argument lists whose structure mirrors that of the component. The techniques are currently based on the C++ language, although work is under way to extend them to other languages (particularly Java).

---

**Assembling Software Products from Feature Selections**

Arie van Deursen, CWI
Merin de Jonge, CWI
Tobias Kuipers, Software Improvement Group

**Abstract:** In this paper we discuss the construction of software products from customer-specific feature selections. We address variability management with the Feature Description Language (FDL) to capture variation points of product line architectures. We describe feature packaging which covers selecting and packaging implementation components according to feature selections using the autobundle tool. Finally, we discuss a generic approach, based on the abstract factory design pattern, to make instantiated (customer-specific) variability accessible in applications.
The solutions and techniques presented in this paper are based on our experience with the product line architecture of the commercial documentation generator DocGen.

Feature Interaction and Dependencies: Modeling Features for Re-engineering a Legacy Product Line

Stefan Ferber, Robert Bosch Research
Jürgen Haag, Robert Bosch Gasoline Systems
Juha Savolainen, Nokia Research Center

Abstract: Re-engineering a legacy product line has been little addressed by current product line research activities. This paper introduces a method to investigate feature dependencies and interactions, which restricts the variants that can be derived from the legacy product line assets. Reorganizing the product line assets with respect to new requirements needs more knowledge than what is easily provided by the classical feature modeling approaches. Hence, adding all the feature dependencies and interactions into the feature tree results in unreadable and unmanageable feature models, which fail to achieve their original goals.

We therefore propose two complementary views to represent the feature model. One view shows hierarchical refinement of features similar to common feature modeling approaches in a feature tree. The second view describes what kind of dependencies and interactions there are between various features.

We show two examples of feature dependencies and interactions in the context of an engine control software product line and we demonstrate how our approach helps define correct product configurations out of product line variants.

Feature Modeling: a Meta-Model to Enhance Usability and Usefulness

Daniel Fey, Nokia Research Center
Robert Fajta, Nokia Research Center
Andras Boros, Nokia Research Center

Abstract: A feature model captures the stakeholder-visible aspects and characteristics of a product line. By revealing a product line's inherent commonalities and variabilities, it acts as a key driver in the creation of core assets. Usability and usefulness, however, are important qualities for a feature model to possess in order to fulfill its role. In our opinion, these qualities can be ensured by building upon an adequate meta-model. The purpose of this article is to describe an extended meta-model for feature
modeling. Meta-model elements, such as features and inter-feature relations, are presented in detail. We propose automated model analysis as the way of extracting information encapsulated in a feature model: algorithms are suggested for identification of the commonality and variability in the modeled product line and for automated consistency checking of products.

---

**On the Influence of Variabilities on the Application Engineering Process of a Product Family**

Lars Geyer, University of Kaiserslautern
Martin Becker, University of Kaiserslautern

**Abstract**: Product Families typically comprise a set of software assets, which offer the possibility to configure the product family to the needs of a specific application. The configuration process is driven by the variabilities, i.e., the variable requirements that were implemented into the software assets in the form of variation points. During application engineering, a developer selects a consistent set of variabilities, this set is used in order to instantiate the family assets to the needed functionality. The paper describes the influence of this configuration step onto the application engineering process of a product family. In addition, it identifies the requirements imposed onto a configuration technique by the described product family application engineering process.

---

**Representing Variability in Software Product Lines: A Case Study**

Michel Jaring, University of Groningen
Jan Bosch, University of Groningen

**Abstract**: This paper focuses on variability in industrial software product lines. Variability is the ability to change or customize a software system, i.e., software architects anticipate change and design architectures that support those changes. If the architecture is used for different product versions, e.g., in a software product line context, it becomes important to understand where change has to be planned and the options possible in particular situations. Three variability issues have been identified in a case study involving a Dutch software company. In our opinion, this company, Rohill Technologies BV, is representative for small and medium sized enterpises in software industry. The issues identified are discussed and analyzed and illustrate the need for handling variability in a more explicit manner. In this paper, we suggest a method to represent and normalize variability in industrial software systems. The method is exemplified by applying it to Rohill’s software product line.
Governance Polarities of Internal Product Lines

Truman Jolley, Boeing Commercial Airplanes
David Kasik, Boeing Commercial Airplanes
Conrad Kimball, Boeing Commercial Airplanes

Abstract: Tension occurs when multiple organizations develop and deliver their own product lines to a single user community. We apply polarity management to governance of the shared architecture, products, and processes for delivery and management of tens of product lines containing hundreds of applications for thousands of engineering users in Boeing Commercial Airplanes. This paper focuses on the use of polarity management to construct extensible governance bodies and processes for a second phase of product line expansion. We define polarity to be two principles which are both true, but conflict. Polarities are often mistaken to be problems to be solved; however, polarities are held, not solved. Polarity management of the product line infrastructure, a complex customer-supplier network, identifies primary organizational tensions that require management; poorly held polarities cause chaos and failure.

Using a Marketing and Product Plan as a Key Driver for Product Line Asset Development

Kyo Kang, Pohang University of Science and Technology
Patrick Donohoe, Software Engineering Institute
Eunman Koh, Pohang University of Science and Technology
Kwanwoo Lee, Pohang University of Science and Technology
Jaejoon Lee, Pohang University of Science and Technology

Abstract: The product line engineering paradigm has emerged recently to address the needs to minimize the development cost and the time to market in this highly competitive global market. Product line development consists of product line asset development and product development using the assets. Product line requirements are essential inputs to product line asset development. These inputs, although critical, are not sufficient to develop product line assets. A marketing and product plan, which includes plans on what features are to be packaged in products, how these features will be delivered to customers (e.g., feature binding time), and how the products will evolve in the future, also drives product line asset development; thus this paper explores design issues from the marketing perspective and presents key design drivers that are tightly coupled with the marketing strategy. An elevator control software example is used to illustrate how the product line asset development is related with the marketing and product plan.
Method for Product-Line Scoping based on Decision-Making Framework

Tomoji Kishi, NEC Corporation
Natsuko Noda, NEC Corporation
Takuya Katayama, Japan Advanced Institute of Science and Technology

Abstract: It is indispensable for strategic product-line development, to define proper scope of the product-line. Once scope has been defined, we examine product-line architecture for it to examine systematic reuse for the product-line. Therefore, in defining scope, we have to examine whether or no it is appropriate for products in the product-line to share the same architecture. The appropriateness of sharing the same architecture among multiple products has to be examined from two points of view. One is from the point of view of individual optimal, i.e. if it is good for each product to use the shared architecture, and the other is from that of whole optimal, i.e. if it is good for product-line as a whole to share the architecture. In this paper, we propose a method for product-line scoping. We consider scoping as decision-making activities, in which we evaluate multiple candidates of scoping, and select proper one examining the appropriateness from the two points of view. In order to demonstrate the applicability of the method, we applied the method to the scoping for products examined for on-board systems in Japanese ITS (Intelligent Transport Systems) projects.

Engineering Software Architectures, Processes and Platforms for System Families - ESAPS Overview

Frank Van der Linden, Philips Medical Systems

Abstract: Between July 1999 and June 2001, 22 European companies and research institutes worked together in the ESAPS project to enhance their capabilities for engineering software for system families. This paper describes the main objectives of the project, and an overview of the results obtained in the project. Finally, the project is related to other projects and initiatives with similar goals.

Variation Management for Software Production Lines

Charles Krueger, BigLever Software, Inc.

Abstract: Variation management in a software product line is a multi-dimensional configuration management problem. In addition to the conventional configuration management problem of managing variation over time, software product lines also have the problem of managing variation among the the individual products in the domain space. In this paper, we illustrate how to divide and conquer the
variation management problem into a collection of nine smaller problems and solutions. We also show how to address the nine problems with lightweight solutions that can reduce the risks, costs, and time for establishing and maintaining a software product line.

---

**Using First Order Logic for Product Line Model Validation**

Mike Mannion, Glasgow Caledonian University

**Abstract:** Product line models are used to drive the generation of requirements for single systems in the product line. They are difficult to validate because they are large and complex. By modelling variability and dependency between requirements using propositional connectives a logical expression can be developed for the model. Validation of the selection of requirements from the model can be achieved by satisfying the logical expression. This approach can be used to validate the model as a whole. A detailed worked example is presented and the computational aspects of the approach are discussed.

---

**Model-driven Product Line Architectures**

Dirk Muthig, Fraunhofer IESE
Colin Atkinson, Fraunhofer IESE

**Abstract:** It has long been recognized that successful product line engineering revolves around the creation of a coherent and flexible product line architecture which consolidates the common parts of a product family for reuse and captures the variant parts for simple adaptation. However, it has been less clear what form such architectures should take and how they should be represented. One promising approach is offered by the new Model Driven Architecture (MDA) paradigm of the OMG, which holds that an organization's key architectural assets should be represented in an abstract "platform independent" way, in terms of UML models, and thereby shielded from the idiosyncrasies and volatility of specific implementation technologies. In this paper we discuss the opportunities and challenges involved in using the MDA paradigm for product line engineering, and explain how model driven, product line architectures can be developed, maintained and applied. After first outlining the core concepts of product line engineering and ad hoc strategies currently used to support it, the paper provides a detailed metamodel of the information that needs to be stored within a product-line architecture. It then explains how these concepts can be added to the existing UML metamodel in a clean and coherent way, thereby extending the language to support product-line engineering.

---

**Widening the Scope of Software Product Lines - From Variation to**

---
Composition

Rob van Ommering, Philips Research Laboratories
Jan Bosch, University of Groningen, The Netherlands

Abstract: Architecture, components and reuse form the key elements to build a large variety of complex, high-quality products with a short lead-time. But the balance between an architecture-driven and a component-driven approach is influenced by the scope of the product line and the characteristics of the development organization. This paper discusses this balance and claims that a paradigm shift from variation to composition is necessary to cope with an increasing diversity of products created by an ever-larger part of an organization. We illustrate our claim with various examples.

Product line annotations with UML-F

Wolfgang Pree, University of California, Berkeley
Marcus Fontoura, IBM Almaden Research Center
Bernhard Rumpe, Munich University of Technology

Abstract: The Unified Modeling Language (UML) community has started to define so-called profiles in order to better suit the needs of specific domains or settings. Product lines represent a special breed of systems they are extensible semi-finished pieces of software. Completing the semi-finished software leads to various software pieces, typically specific applications, that share the same core. Though product lines have been developed for a wide range of domains, they apply common construction principles. The intention of the UML-F profile is the definition of a UML subset, enriched with a few UML-compliant extensions, that allows the annotation of such artefacts. The paper presents aspects of the profile with a focus on patterns and exemplifies its usage.

Evolutionary Introduction of Software Product Lines

Daniel Simon, Universität Stuttgart
Thomas Eisenbarth, Universität Stuttgart

Abstract: Software product lines have proved to be a successful and efficient means for managing the development of software in industry. The significant benefits over traditional software architectures have the potential to convince software companies to adopt the product line approach for their existing products. In that case, the question arises how to best convert the existing products into a software product line. For several reasons, an evolutionary approach is desirable. But so far, there is little
guidance on the evolutionary introduction of software product lines.

In this paper, we propose a lightweight iterative process supporting the incremental introduction of product line concepts for existing software products. Starting with the analysis of the legacy code, we assess what parts of the software can be restructured for product line needs at reasonable costs. For the analysis of the products, we use feature analysis, a reengineering technique tailored to the specific needs of the initiation of software product lines.

---

**Using Options Analysis for Reengineering (OAR) for Mining Components for a Product Line**

Dennis Smith, Software Engineering Institute
Liam O'Brien, Software Engineering Institute
John Bergey, Software Engineering Institute

**Abstract:** Options Analysis for Reengineering (OAR) is a systematic, architecture-centric means for mining existing components for a product line or new software architecture. The method incorporates a set of scalable techniques and exercises to collaboratively analyze existing components, determine viable mining options, and evaluate the most promising options. OAR has 5 activities which are followed in a systematic manner to identify components for mining and estimate the cost and risk of changes required to each legacy component to enable its reuse within a new software architecture. OAR provides visibility into this highly complex analysis activity. It also provides insights into implicit stakeholder assumptions, constraints, and other major drivers that impact the mining of components. Results from a pilot application of OAR are presented in this paper.

---

**Systematic Integration of Variability into Product Line Architecture Design**

Steffen Thiel, Robert Bosch GmbH
Andreas Hein, Robert Bosch GmbH

**Abstract:** Product lines consider related products, their commonalities and their differences. The differences between the single products are also referred to as variability. Consequently, variability is inherent in every product line and makes the key difference compared to single systems. While on the requirements level the methods for analyzing product line variability are understood today, its transition to architecture remains vague. Bringing variability to architecture as an add-on is just a provisional solution and forebodes the risk of violating other intentions. This paper presents a systematic approach to integrating variability with product line architecture design. In particular, it promotes variability as an architectural driver, embeds variability requirements in the architecture design framework QUASAR,
and gives guidelines and examples for documenting variability in architectural views.

---

**Critical Factors for a Successful Platform-based Product Family Approach**

Jan Gerben Wijnstra, Philips Research Laboratories

**Abstract:** The idea of software product families is becoming more and more popular, both in research and in industry. When listening to the ideal story, the benefits of product families are stressed and very little attention is paid to possible problems. However, in practice, it is important to know what these problems are and how they can be dealt with. In this paper we want to identify some of the most critical issues, and how the can be handled. This will be done based on the experiences from three industrial product families for professional markets. The focus will be on product families that use platforms with reusable assets for the development of the family members.

---

**Establishing a Software Product Line in an Immature Domain**

Stefan Voget, Robert Bosch GmbH
Martin Becker, University of Kaiserslautern

**Abstract:** Often product lines are applied to stable domains, i.e. a set of common features is identifiable in advance and the evolution of the domain is manageable during lifetime. These prerequisites are not always given. But there may be a market pressure which requires to develop products with systematic and preplanned reuse in a domain not properly overlookable. In such a case the product line approach also offers a set of methods which helps to overcome the risks of an immature domain. In this paper we consider such risks and present some approaches to manage them. The considerations are substantiated by experiences made in the domain of driver information systems in an automotive context. The development is deeply influenced by technological changes (e.g. Internet, MP3-player, UMTS) that challenge the successful deployment of product line technology.

---

**Performance Analysis of Component-based Applications**

Sherif Yacoub, Hewlett-Packard Laboratories

**Abstract:** Performance analysis is a software engineering activity that involves analyzing a software application with respect to performance quality attributes such as response and execution times.
Performance analysis tools provide the necessary support for the analyst to monitor program execution, record and analyze performance data, and locate and understand areas of poor performance. Performance analysis methods and techniques are highly dependent on the properties of the software system to be analyzed. Product line engineering applications possess some special properties that impose constraints on the selection of the performance analysis techniques to be applied and the tools to be used. The development of a component-based reference architecture is crucial to the success of a true product line. The component-based nature facilitates the integration of components and the replacement of a component with another to meet the requirements of an instance application of the product line. In this paper, we discuss performance analysis of component-based software systems and its automation. We discuss how component-based system properties influence the selection of methods and tools used to obtain and analyze performance measures. We use a case study of document content re-mastering product line to illustrate the application of a performance analysis method to component-based applications.

Product Line Architecture and the Separation of Concerns

Jay van Zyl, Rubico Products (Pty) Ltd

Abstract: Software product lines present many benefits over the traditional methods of building systems. With the diverse implementation of both application and technology architectures, organizations are faced with complex design constraints. Layered architectures assist with breaking down complexity through separating elements based on their use and applicability. To really achieve high levels of re-use and productivity without only focusing on one architectural style, is difficult to do. This paper discusses two primary concepts namely, product line architecture and separation continuum. It starts by presenting a product line architectural view that shows how various concepts are separated based on abstraction. In order to provide context, the Software Engineering Institute and Carnegie Mellon University’s product line practices are briefly discussed. The separation continuum shows how vertical and horizontal layering can assist with separating user interface from business logic and data, also the separation of customer facing processes from infrastructure facing processes. Software product developers know that these relationships are not easily related. Customer facing business processes have different requirements to infrastructure facing processes. In order to tie all the concepts together, vertical layering is needed whereby the more abstract elements are separated from the implementation of those elements. An application assembly approach is discussed whereby the product line architecture is tied to the separation continuum showing how high levels of productivity can be achieved when realizing product lines. The approach presented in this paper is still under development with implementation on a limited number of product lines only. It is intended that the content will provoke and stimulate the thinking and experimentation needed to deal with application assembly by means of having a separation continuum and a matching product line architecture.
The Second Software Product Line Conference

Birds-of-a-Feather Sessions

SPLC2 will provide a forum for people to meet informally and discuss product-line-related issues in a relaxed setting. These birds-of-a-feather sessions will occur the evening of Wednesday, August 21. Any SPLC2 participant may create a birds-of-a-feather session. Simply post a description for the session on the conference's message board, and choose a time and place from the available meeting rooms (which will also be posted on the message board). Rooms will be allocated on a first-come-first-served basis.

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

Copyright ©2002 by Carnegie Mellon University