

# MVDC 2: Managing Variabilities consistently in Design and Code

OOPSLA 2005 workshop proposal

## ***Themes and goals:***

Software product lines / software families promise a reduction of development time and product development cost through systematic reuse and improved software quality. The idea is to develop the common parts of a family just once while being prepared to adapt these parts to the different product requirements in a controlled way.

With the introduction of software product line approaches into the practice, variants and variability add a new dimension of complexity to the software development process. The combinatorial explosion of possible variants in systems with a high degree of variability requires special concepts for specifying, modeling, and implementing these systems to assure quality and functionality.

Object technology encompasses a diversity of variability approaches at different abstraction levels. During the analysis phase, OO provides useful techniques such as use cases or scenarios. During design, patterns, and frameworks serve to handle variability. At the implementation level, OO supports variability by, for example, dynamic binding, polymorphism, inheritance and sub-typing.

These approaches are not integrated and none of them explicitly supports the description and implementation of variants.

Feature models are better suited for planning and supporting strategic decision both for architectural and for component development. Feature models are tightly connected to object-oriented models of the UML. Feature Models are used in different ways:

- \* They are the basis for choosing a set of features by a customer in order to configure a product, especially by expressing feature dependencies.
- \* They support the navigation in requirements specification.
- \* They are used for directing and coordinating the partitioning of architecture and implementation according to the Separation of Concerns principle.
- \* They are applied for composing the final software product.

For all the above mentioned reasons, Feature Models play a central role in Product Line development. However, their definition in terms of syntax, semantics and graphical representation as well as the integration into the software development process is still limited.

Likewise, OO languages do not give much support in making variants explicit. AO languages seem to be more appropriate for implementing a feature in a separate concern and making features first class citizens within the software.

The goal of this workshop is to evaluate existing technologies for their ability to manage variants and the resulting variability in design and code consistently, identify gaps and find points for collaboration to fill these gaps.

This workshop continues the work we started in the MVCDC workshop at OOPSLA 2004.

### **Organizing committee:**

- **Christa Schwanninger (primary contact)**  
Siemens AG  
CT SE 2  
Otto-Hahn-Ring 6  
81739 Munich  
Germany  
Phone: +49 89 63649477  
Fax: + 49 89 63645450  
e-mail: [christa.schwanninger@siemens.com](mailto:christa.schwanninger@siemens.com)
- **Danilo Beuche**  
pure-systems GmbH  
Agnetenstr. 14  
39106 Magdeburg  
<http://www.pure-systems.com>  
Phone: +49-391-54456910  
Fax: +49-391-54456990  
e-mail: [danilo.beuche@pure-systems.com](mailto:danilo.beuche@pure-systems.com)
- **Krzysztof Czarnecki**  
University of Waterloo  
Department of Electrical &  
Computer Engineering  
200 University Ave. West  
Waterloo, ON N2L 3G1  
Canada  
Phone: +1 (519) 888-4567 x7137  
Fax: +1 (519) 746-3077  
e-mail: [czarnecki@acm.org](mailto:czarnecki@acm.org)
- **Mira Mezini**  
Darmstadt University of Technology  
Alexanderstr. 10  
64283 Darmstadt  
Germany  
Phone: + 49 6151 165311  
Fax: +49 6151 165410  
e-mail: [mezini@informatik.tu-darmstadt.de](mailto:mezini@informatik.tu-darmstadt.de)
- **Markus Völter**  
ingenieurbüro für softwaretechnologie  
Ziegelaecker 11  
89520 Heidenheim  
Germany  
Phone: . +49 (0) 73 21 / 97 33 44  
e-mail: [voelter@acm.org](mailto:voelter@acm.org)

**Rainer Burgstaller**  
Siemens AG  
Otto-Hahn-Ring 6  
81739 Munich  
Germany

### ***Organizers' background:***

**Danilo Beuche**, is managing director of the pure-systems GmbH. pure-systems is a software company specialized in services and tool development for the application of product line technologies in embedded software systems. When he joined the GMD First (now Fraunhofer FIRST) in 1995, he started to work in the field of embedded operating systems and software families and continued this line of work at the University Magdeburg. His work on tool support for feature based software development has gained interest both in the research community and in the industry and finally lead to the founding of pure-systems in 2001.

**Krzysztof Czarnecki** is an Assistant Professor at the University of Waterloo, Canada. Before coming to Waterloo, he spent 8 years at DaimlerChrysler Research working on the practical applications of generative programming. He is co-author of the book "Generative Programming" (Addison-Wesley, 2000), which is regarded as founding work of the area and is used as a graduate text at universities around the world. He was General Chair of the 2003 International Conference on Generative Programming and Component Engineering (GPCE) and will deliver a keynote on generative software development at the at the UML 2004 conference in Lisbon. His current research focuses on realizing the synergies between generative programming and MDA.

**Mira Mezini** is professor of computer science at Darmstadt University of Technology, where she is leading the software technology group. Her research focuses on developing methods and tools that support programmers in writing software that has a clear modular structure that reflects a clear separation of concerns. A key theme underlying her current work is aspect-oriented software development (AOSD). She is leading several important projects in this area.

**Christa Schwanninger**, Senior Research Scientist at Siemens AG, Corporate Technology, Munich, Germany. Her fields of interest are software architecture, distributed object computing, patterns, frameworks and aspect oriented software development. She participates in industrial research in new and promising areas of software engineering and is a consultant for business units that actually build commercial software, mostly embedded in small, resource restricted devices. She was conference chair of EuroPLoP 2001 and 2002, was/is member of the program committee of EuroPLoP 2000 and 2003 and has (co) organized several workshops and tutorials before. Among them are the Pattern Writing Workshops at two EuroPLoP conferences (1999, 2000) and a series of pattern writing tutorials at OOPSLA 98, OOPSLA 99, Software Developers Conference in San Francisco '99 and SIGS Application Development '99. She co-organized a workshop on Deploying Lightweight Processes at OOPSLA 2000, a workshop on patterns and aspects

("Beyond Design: Patterns(mis)used") at OOPSLA 2001 and a workshop on reuse in constrained environments at OOPSLA 2003.

**Markus Voelter**, Independent Consultant, Heidenheim, Germany

Markus Völter works as a freelance consultant for software technology and engineering. He focuses on the architecture of large, distributed systems. His interests include patterns, frameworks, components, middleware as well as generative and model-driven development. Markus is the author of various technical articles and papers as well as several published patterns. He is a regular speaker at national and international conferences and co-author of Wiley's "Server Component Patterns" book. Over the last years, Markus has worked on several projects of different sizes in different domains such as banking, media, astrophysics and automotive. Most recently, he has been working on the architecture of embedded software, specifically the small components project, which aims at providing component infrastructures for embedded systems. Markus holds a Diploma in Technical Physics. He can be reached at voelter@acm.org or www.voelter.de

### ***Previous Workshops***

Christa Schwanninger, Danilo Beuche, Krzysztof Czarnecki, Mira Mezine, Markus Völter, Workshop - MVCDC: Managing Variabilities consistently in Design and Code, OOPSLA '04 conference, Vancouver, Canada, October 2004

Markus Voelter, Michael Kircher, Christa Schwanninger, Uwe Zdun, Alexander Schmid, Workshop - Reuse in Constrained Environments, OOPSLA'03 conference, Anaheim, California, USA , October 2003

Matthias Riebisch, Jim Coplien, Detlef Streitferdt, Workshop on Modeling Variability for Object-Oriented Product Lines, ECOOP'03, Darmstadt, July 2003

Bedir Tekinerdogan, Mehmet Aksit, Krzysztof Czarnecki, Sholom Cohen , OOPSLA'01 Workshop on Managing Variability in Domain Engineering using OO Technology, OOPSLA'01 conference, Tampa, Florida, USA, October 2001

### ***Special requirements***

3 Flip charts with board markers.

### ***Expected number of participants***

The maximum number of participants is 15. To ensure a minimal number of participants, we will use the following means:

- We know many people in the community that are interested.
- We will advertise the workshop on our homepages.
- We will publish CFPs on various applicable mailing lists (such as AOSD\_announce, ECOOP list, SE World, several generative programming lists).

### ***Pre-workshop activities***

On May 27, (a week after potential acceptance of the workshop) we will have an initial web page available with:

- introduction into workshop topic
- workshop description
- submission guidelines

- deadlines
- workshop format description
- introduction of organizers

When the deadline has passed:

- program committee evaluates submissions
- notification of acceptance according to OOPSLA deadlines
- all accepted papers published after notification of acceptance

Two weeks before workshop: a detailed workshop schedule will be published together with what every participant is expected to do to prepare for the workshop.

### ***Workshop preparation***

Every interested person is invited to apply for attendance by sending a position paper to the organizers. The submission should be one to two pages describing the key ideas. Submissions will be reviewed by the organizers. The authors will be notified about acceptance before the early registration deadline.

### ***Planned workshop activities***

The workshop is laid out as full day workshop.

Instead of presenting the papers, each participant will be expected to review everyone else's paper before the workshop and complete the following two sentences for each:

- What I really liked about this paper is ...
- The most important question I would like to ask the author is ...

These answers are written down on index cards and will be collected before the workshop. During the workshop, we will spend the morning with questions and answers to gain deeper insight into the problem described in the paper. Before each paper session, the author will be permitted a 4 minute slot to very briefly present his paper/work. The workshop format in the afternoon will be utilizing the "Open Space" format in order to discuss topics of interest that might be, or might not be directly related to the papers presented in the morning.

### ***Post-workshop activities***

The workshop results are published on the MVCDC Wiki at

<http://catamaran.labs.cs.uu.nl/twiki/pt/bin/view/Variability/WorkshopManagingVariabilityConsistentlyInDesignAndCode>, extending the results from the 2004 MVCDC

workshop.. We will provide a summary of the workshop and aim on a joint research agenda for investigating ways to handle variants consistently in design and code.

If we receive a significant amount and quality of submissions, we also consider trying a special issue of some Journal.

### ***Abstract***

With the introduction of software product line approaches into the practice, variants and variability add a new dimension of complexity to the software development process. The combinatorial explosion of possible variants in systems with a high degree of variability requires improved and changed concepts for specifying, modeling, and implementing these systems to assure quality and functionality.

Current OO technologies and standardized design languages fall short in offering appropriate support for designing and implementing variabilities. Additionally the

integration of design tools and implementation technologies concerning variants is still poor.

The main goal is to discuss and identify efficient ways for dealing with highly variable software systems on design and code level by evaluating existing approaches and new ideas from the research community and industrial practice.

Potential topics include, but are not limited to:

- tools and processes for managing variants
- modeling languages for capturing variants
- code generation support
- feature oriented development tools and technologies
- languages for independent feature development and feature composition
- AO approaches for independent feature development and feature composition
- Integration of variant management into model driven software development techniques
- Support for model to model and model to code transformations