

*Commenced Publication in 1973*

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

## Editorial Board

David Hutchison

*Lancaster University, UK*

Takeo Kanade

*Carnegie Mellon University, Pittsburgh, PA, USA*

Josef Kittler

*University of Surrey, Guildford, UK*

Jon M. Kleinberg

*Cornell University, Ithaca, NY, USA*

Friedemann Mattern

*ETH Zurich, Switzerland*

John C. Mitchell

*Stanford University, CA, USA*

Moni Naor

*Weizmann Institute of Science, Rehovot, Israel*

Oscar Nierstrasz

*University of Bern, Switzerland*

C. Pandu Rangan

*Indian Institute of Technology, Madras, India*

Bernhard Steffen

*University of Dortmund, Germany*

Madhu Sudan

*Massachusetts Institute of Technology, MA, USA*

Demetri Terzopoulos

*New York University, NY, USA*

Doug Tygar

*University of California, Berkeley, CA, USA*

Moshe Y. Vardi

*Rice University, Houston, TX, USA*

Gerhard Weikum

*Max-Planck Institute of Computer Science, Saarbruecken, Germany*

Gabor Karsai Eelco Visser (Eds.)

# Generative Programming and Component Engineering

Third International Conference, GPCE 2004  
Vancouver, Canada, October 24-28, 2004  
Proceedings

## Volume Editors

Gabor Karsai  
Vanderbilt University  
Institute for Software Integrated Systems (ISIS)  
Nashville, TN 37235, USA  
E-mail: gabor.karsai@vanderbilt.edu

Eelco Visser  
Universiteit Utrecht  
Institute of Information and Computing Sciences  
P.O. Box 80089, 3508 TB Utrecht, The Netherlands  
E-mail: visser@acm.org

Library of Congress Control Number: 2004113646

CR Subject Classification (1998): D.2, D.1, D.3, K.6

ISSN 0302-9743  
ISBN 3-540-23580-9 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media  
springeronline.com

© Springer-Verlag Berlin Heidelberg 2004  
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Olgun Computergrafik  
Printed on acid-free paper      SPIN: 11338000      06/3142      5 4 3 2 1 0

# Preface

Generative and component approaches have the potential to revolutionize software development in a similar way as automation and components revolutionized manufacturing. Generative Programming (developing programs that synthesize other programs), Component Engineering (raising the level of modularization and analysis in application design), and Domain-Specific Languages (elevating program specifications to compact domain-specific notations that are easier to write and maintain) are key technologies for automating program development.

GPCE arose as a joint conference, merging the prior conference on Generative and Component-Based Software Engineering (GCSE) and the Workshop on Semantics, Applications, and Implementation of Program Generation (SAIG). The goal of GPCE is to provide a meeting place for researchers and practitioners interested in cutting edge approaches to software development. We aim to foster further cross-fertilization between the software engineering research community on the one hand, and the programming languages community on the other, in addition to supporting the original research goals of both the GCSE and the SAIG communities.

This volume contains the proceedings of the *Third International Conference on Generative Programming and Component Engineering*, held in Vancouver, Canada from October 24 to 28, 2004, where it was co-located with OOPSLA 2004 and ISSM 2004.

Responding to the call for papers 89 abstracts were submitted, 75 of which materialized as full paper submissions. The papers were reviewed by program committee members and their co-reviewers who together produced a total of 250 reviews, between 3 and 5 per paper. Reviews were often thorough and sometimes actually included the views of multiple co-reviewers. Consensus about the papers to be accepted was reached during the online program committee meeting held in the second week of May 2004. The meeting consisted of a discussion by email of each of the papers by the entire PC so that members could get an overall impression of the quality of the submitted papers, beyond the ones they reviewed themselves. The committee selected 25 of the 75 papers for presentation at the conference and publication in the proceedings. Of the accepted papers, two are co-authored by PC members (from a total of six PC submissions). We tried hard to ensure fairness and held PC submissions to a high standard. Paper submission and reviewing were supported by the open source version of the CyberChair conference system installed at the webserver of the Institute of Information and Computing Sciences of Utrecht University, The Netherlands.

In addition to the technical paper presentations the conferences featured two invited speakers, a panel, four tutorials, five workshops, and six demonstrations.

**Invited Speakers.** The keynote talk by Jack Greenfield examined the *software factory* approach to rapidly develop domain-specific languages and tools to auto-

mate the production of applications in specific domains, combining innovations in adaptive assembly, software product lines, and model driven development.

The invited talk by Peter Mosses gave an overview of the state of the art in *modular language description*, i.e. the specification of the semantics of programming language features in separate modules such that new languages can be defined by module composition.

**Panel.** A panel chaired by Gabor Karsai and further consisting of Don Batory, Krzysztof Czarnecki, Jeff Gray, Douglas Schmidt, and Walid Taha examined the current state of the field of generative programming, addressing issues such as its relevance for information technology practice, incorporating generative approaches in education, evaluation and comparison of generative technologies, and research challenges.

**Tutorials.** The four GPCE tutorials gave introductions to important areas of the generative programming field:

- *Adaptive object-model architecture: Dynamically adapting to changing requirements* by Joe Yoder
- *Multi-stage programming in MetaOCaml* by Walid Taha and Cristiano Calcagno
- *Generative software development* by Krzysztof Czarnecki and Jack Greenfield
- *Program transformation systems: Theory and practice for software generation, maintenance, and reengineering* by Ira Baxter and Hongjun Zheng

**Workshops.** Prior to GPCE 2004 six workshops were held, providing an opportunity for attendees to exchange views in subareas of generative programming.

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. In the **OOPSLA/GPCE Workshop on Managing Variabilities Consistently in Design and Code** participants discussed and identified 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.

The **Software Transformation Systems Workshop** was designed to investigate the use of software transformation tools to support generative programming by looking at various generative techniques and suggesting how these may be supported by various general purpose transformation tools, leading to a more general understanding of common principles for supporting generative methods.

MetaOCaml is a multi-stage extension of the widely used functional programming language OCaml. It provides a generic core for expressing macros, staging, and partial evaluation. The **First MetaOCaml Workshop** provided a forum for discussing experience with using MetaOCaml as well as possible future developments for the language.

The **6th GPCE Young Researchers Workshop** provided a platform for young international researchers to present their work and receive feedback from experienced panelists.

The **OOPSLA/GPCE Workshop on Best Practices for Model-Driven Software Development** brought together practitioners, researchers, academics, and students to discuss the best practices for the development of model-driven software, and to discuss the state of the art of tool support for MDSD, including emerging Open Source tool products for model-driven development of software systems.

**Demonstrations.** The following demonstrations were held in parallel to the technical paper program:

- *Implementation of DSLs using staged interpreters in MetaOCaml* by Kedar Swadi from Rice University
- *MetaEdit+: Domain-specific modeling for full code generation demonstrated* by Juha-Pekka Tolvanen from MetaCase
- *Towards domain-driven development: the SmartTools software factory* by Didier Parigot from INRIA Sophia-Antipolis
- *Xirc: Cross-artifact information retrieval* by Michael Eichberg, and Thorsten Schaefer from Darmstadt University of Technology
- *C-SAW and GenAWeave: A two-level aspect weaving toolsuite* by Jeff Gray, Jing Zhang, and Suman Roychoudhury, from the University of Alabama at Birmingham and Ira Baxter from Semantic Designs
- *The concern manipulation environment* by Peri Tarr, Matthew Chapman, William Chung, and Andy Clement, from the IBM Thomas J. Watson Research Center and IBM Hursley Park.
- *Program transformations for re-engineering C++ components* by Ira Baxter, Larry Akers, Semantic Designs, and Michael Mehlich from Semantic Designs.

The program of this year's conference is proof that the GPCE community is a vibrant, lively group that produces significant new contributions.

# Organization

GPCE 2004 was organized by the Association for Computing Machinery (ACM), the OGI School of Science & Engineering at OHSU (USA), Utrecht University (The Netherlands), Vanderbilt University (USA), Intel (USA), University of Alabama at Birmingham (USA), and the University of Waterloo (Canada). The event was sponsored by ACM SIGPLAN, ACM SIGSOFT, and Microsoft and co-located with OOPSLA 2004 and ISSM 2004 in Vancouver, Canada.

## General Chair

Tim Sheard, OGI School of Science & Engineering at OHSU, Portland, Oregon, USA

## Program Chairs

Gabor Karsai, Vanderbilt University, USA  
Eelco Visser, Utrecht University, The Netherlands

## Program Committee

Uwe Aßmann (Linköpings Universitet, Sweden)  
Don Batory (University of Texas, USA)  
Jan Bosch (Universiteit Groningen, The Netherlands)  
Jean Bezivin (Université de Nantes, France)  
Jim Cordy (Queen's University, Canada)  
Krzysztof Czarnecki (University of Waterloo, Canada)  
Mathew Platt (University of Utah, USA)  
Robert Glück (Københavns Universitet, Denmark)  
George Heineman (Worcester Polytechnic Institute, USA)  
Michael Leuschel (University of Southampton, UK)  
Karl Lieberherr (Northeastern University, USA)  
Simon Peyton Jones (Microsoft Research, UK)  
Douglas R. Smith (Kestrel Institute, USA)  
Gabriele Taentzer (Technische Universität Berlin, Germany)  
Todd Veldhuizen (Indiana University, USA)  
Kris de Volder (University of British Columbia, Canada)  
Dave Wile (Teknowledge Corporation, USA)  
Alexander Wolf (University of Colorado at Boulder, USA)

## Workshop Chair

Zino Benaissa, Intel, USA

## Tutorial Chair

Jeff Gray, University of Alabama at Birmingham, USA

## Demonstrations Committee

Simon Helsen (chair), University of Waterloo, Canada

William Cook, University of Texas at Austin, USA

Frédéric Jouault, Université de Nantes, France

## Co-reviewers

Nils Andersen	Yvonne Howard	Jeffrey Palm
Anya H. Bagge	Anton Jansen	Emir Pasalic
Ivor Bosloper	Michel Jaring	Andrew Pitts
Martin Bravenboer	Dean Jin	Stephane Lo Presti
Niels H. Christensen	Merijn de Jonge	Armin Rigo
Alessandro Coglio	Frederic Jouault	Matthew Rutherford
S. Deelstra	Kazuhiko Kakehi	Kevin A. Schneider
Juergen Dingel	Karl Trygve Kalleberg	Jörg Schneider
Eelco Dolstra	Markus Klein	Ulrik Pagh Schultz
Alexander Egyed	Andrei Klimov	Johanneke Siljee
Dan Elphick	Jia Liu	M. Sinnema
Lindsay Errington	Roberto Lopez-Herrejon	Therapon Skotiniotis
Natalya Filatkina	David Lorenz	Mike Sperber
Murdoch Gabbay	Andrew Malton	Walid Taha
Hugh Glaser	Katharina Mehner	Edward Turner
Andy Gravell	Anne-Francoise Le Meur	Mauricio Varea
Jurriaan Hage	Torben Mogensen	Andrzej Wasowski
Jan Heering	Dirk Muthig	Andreas Winter
Andre van der Hoek	Karina Olmos	Pengcheng Wu
Kathrin Hoffmann	Scott Owens	

## Steering Committee

Don Batory (University of Texas)

Eugenio Moggi (Università di Genova)

Greg Morrisett (Cornell)

Janos Sztipanovits (Vanderbilt University School of Engineering)

Krzysztof Czarnecki (University of Waterloo)

Walid Taha (Rice University)

Bogdan Franczyk (Universität Leipzig)

Ulrich Eisenecker (Fachhochschule Kaiserslautern)



## Previous Events

GPCE emerged as the unification of the SAIG workshop series and the GCSE conference series.

GPCE 2003, Erfurt, Germany

GPCE 2002, Pittsburgh, Pennsylvania, USA

GCSE 2001, Erfurt, Germany

SAIG 2001, Firenze, Italy

GCSE 2000, Erfurt, Germany

SAIG 2000, Montréal, Canada

GCSE 1999, Erfurt, Germany

See also the permanent website of the conference series <http://www.gpce.org>

## Sponsors



ACM SIGPLAN and ACM SIGSOFT

***Microsoft***<sup>®</sup>

# Table of Contents

## Aspect Orientation

Generating AspectJ Programs with Meta-AspectJ . . . . .	1
<i>David Zook, Shan Shan Huang, and Yannis Smaragdakis</i>	
Splice: Aspects That Analyze Programs . . . . .	19
<i>Sean McDirmid and Wilson C. Hsieh</i>	
A Generative Approach to Aspect-Oriented Programming . . . . .	39
<i>Douglas R. Smith</i>	
Generic Advice: On the Combination of AOP with Generative Programming in AspectC++ . . . . .	55
<i>Daniel Lohmann, Georg Blaschke, and Olaf Spinczyk</i>	
Supporting Flexible Object Database Evolution with Aspects . . . . .	75
<i>Awais Rashid and Nicholas Leidenfrost</i>	
A Pointcut Language for Control-Flow . . . . .	95
<i>Rémi Douence and Luc Teboul</i>	
SourceWeave.NET: Cross-Language Aspect-Oriented Programming . . . . .	115
<i>Andrew Jackson and Siobhán Clarke</i>	

## Staged Programming

Meta-programming with Typed Object-Language Representations . . . . .	136
<i>Emir Pašalić and Nathan Linger</i>	
Metaphor: A Multi-stage, Object-Oriented Programming Language . . . . .	168
<i>Gregory Neverov and Paul Roe</i>	
Optimising Embedded DSLs Using Template Haskell . . . . .	186
<i>Sean Seefried, Manuel Chakravarty, and Gabriele Keller</i>	

## Types of Meta-programming

A Fresh Calculus for Name Management . . . . .	206
<i>Davide Ancona and Eugenio Moggi</i>	
Taming Macros . . . . .	225
<i>Ryan Culpepper and Matthias Felleisen</i>	
A Unification of Inheritance and Automatic Program Specialization . . . . .	244
<i>Ulrik P. Schultz</i>	

## Meta-programming

- Towards a General Template Introspection Library ..... 266  
*István Zólyomi and Zoltán Porkoláb*
- Declaring and Enforcing Dependencies Between .NET Custom Attributes . 283  
*Vasian Cepa and Mira Mezini*
- Towards Generation of Efficient Transformations ..... 298  
*Attila Vizhanyo, Aditya Agrawal, and Feng Shi*

## Model-Driven Approaches

- Compiling Process Graphs into Executable Code ..... 317  
*Rainer Hauser and Jana Koehler*
- Model-Driven Configuration and Deployment  
of Component Middleware Publish/Subscribe Services ..... 337  
*George Edwards, Gan Deng, Douglas C. Schmidt, Aniruddha Gokhale,  
and Bala Natarajan*
- Model-Driven Program Transformation of a Large Avionics Framework ... 361  
*Jeff Gray, Jing Zhang, Yuehua Lin, Suman Roychoudhury, Hui Wu,  
Rajesh Sudarsan, Aniruddha Gokhale, Sandeep Neema, Feng Shi,  
and Ted Bapty*

## Product Lines

- Automatic Remodularization and Optimized Synthesis  
of Product-Families ..... 379  
*Jia Liu and Don Batory*
- VS-Gen: A Case Study of a Product Line for Versioning Systems ..... 396  
*Jernej Kove and Christian Gebauer*
- A Model-Driven Approach for Smart Card Configuration ..... 416  
*Stéphane Bonnet, Olivier Potonniée, Raphaël Marvie,  
and Jean-Marc Geib*

## Domain-Specific Languages and Generation

- On Designing a Target-Independent DSL  
for Safe OS Process-Scheduling Components ..... 436  
*Julia L. Lawall, Anne-Françoise Le Meur, and Gilles Muller*
- A Generative Framework for Managed Services ..... 456  
*Liam Peyton and Arif Rajwani*
- A Generative Approach to the Implementation of Language Bindings  
for the Document Object Model ..... 469  
*Luca Padovani, Claudio Sacerdoti Coen, and Stefano Zacchiroli*

**Invited Speakers**

Software Factories: Assembling Applications  
with Patterns, Models, Frameworks and Tools ..... 488  
    *Jack Greenfield*

Modular Language Descriptions..... 489  
    *Peter D. Mosses*

**Author Index** ..... 491