

PROCEEDINGS OF THE  
IFIP TC2 WORKING CONFERENCE ON  
GENERIC PROGRAMMING

Edited by

JEREMY GIBBONS

University of Oxford

JOHAN JEURING

Utrecht University

**Kluwer Academic Publishers**  
Boston/Dordrecht/London

## Contents

Preface	vii
Acknowledgements	ix
Generic Programming within Dependently Typed Programming <i>Thorsten Altenkirch, Conor McBride</i>	1
Generic Haskell, Specifically <i>Dave Clarke, Andres Löh</i>	21
Generic Accumulations <i>Alberto Pardo</i>	49
A Generic Algorithm for Minimum Chain Partitioning <i>Sharon A. Curtis</i>	79
Concrete Generic Functionals <i>Raymond T. Boute</i>	89
Discussion: The Rôle of Types in Generic Programming	121
Making the Usage of STL Safe <i>Douglas Gregor, Sibylle Schupp</i>	127
Static Data Structures <i>Michael Burton, William Griswold, Andrew McCulloch, Gary Huber</i>	141
Adaptive Extensions of Object-Oriented Systems <i>Olaf Seng, Thomas Gensler, Benedikt Schulz</i>	171
Complete Traversals as General Iteration Patterns <i>William Klostermeyer, David Musser, Arturo Sánchez-Ruíz</i>	187
Efficient Implementation of Run-time Generic Types for Java <i>Eric Allen, Robert Cartwright, Brian Stoler</i>	207
Discussion: Views of Genericity	237

## Preface

Generic programming is a matter of making programs more adaptable by making them more general. Generic programs often embody non-traditional kinds of polymorphism; ordinary programs are obtained from them by suitably instantiating their parameters. In contrast with normal programs, the parameters of a generic programs are often quite rich in structure. For example they may be other programs, types or type constructors, class hierarchies, or even programming paradigms.

Generic programming techniques have always been of interest, both to practitioners and to theoreticians, but only recently have generic programming techniques become a specific focus of research in the functional and object-oriented programming language communities. The IFIP TC2 Working Conference on Generic Programming, held at Schloß Dagstuhl, Germany, on 11th and 12th July 2002, brought together leading researchers in generic programming from around the world, and featured papers capturing the state of the art in this important emerging area.

The conference was sponsored by IFIP Technical Committee 2, and organized in cooperation with Working Group 2.1 on Algorithmic Languages and Calculi.

This book contains revised versions of the papers that were presented at the conference. The papers are recorded in order of presentation. The editors are grateful to the members of the Programme Committee and the reviewers for their careful work, and particularly to Bernhard Möller for also making the local arrangements for the conference. They would also like to thank Lambert Meertens, chairman of WG2.1, for proposing the conference and providing useful guidance and inspiration, Tim Sheard for his elegant definition of generic programming (quoted above), and Yana Lambert of Kluwer Academic Publishers for general support in the preparation of these proceedings. Financial support for the conference from the Deutsche Forschungsgemeinschaft is gratefully acknowledged.

JEREMY GIBBONS  
JOHAN JEURING

## Acknowledgements

### Programme Committee

Matt Austern	Bernhard Möller
Erke Boiten	Oege de Moor
Ulrich W. Eisenecker	David Musser
Jeremy Gibbons (co-chair)	Martin Odersky
Ralf Hinze	Ross Paterson
Johan Jeuring (co-chair)	Simon Peyton Jones
Gary T. Leavens	Colin Runciman
Karl Lieberherr	Doaitse Swierstra
Lambert Meertens	Stephanie Weirich
Eugenio Moggi	

### Reviewers

Matt Austern	Bernhard Möller
Andreas Behrend	Oege de Moor
Erke Boiten	David Musser
Dave Clarke	Martin Odersky
Ulrich W. Eisenecker	Ross Paterson
Jeremy Gibbons	Dusko Pavlovic
Ralf Hinze	Simon Peyton Jones
Johan Jeuring	Christine Röckl
Gary T. Leavens	Colin Runciman
Karl Lieberherr	Doaitse Swierstra
Lambert Meertens	Stephanie Weirich
Andres Löh	Matthias Zenger
Eugenio Moggi	