

# **Systems Implementation 2000**

**Come and visit the  
IT & Applied Computing Resource Centre:  
[www.IT-CH.com](http://www.IT-CH.com)**

## **IFIP – The International Federation for Information Processing**

IFIP was founded in 1960 under the auspices of UNESCO, following the First World Computer Congress held in Paris the previous year. An umbrella organization for societies working in information processing, IFIP's aim is two-fold: to support information processing within its member countries and to encourage technology transfer to developing nations. As its mission statement clearly states,

IFIP's mission is to be the leading, truly international, apolitical organization which encourages and assists in the development, exploitation and application of information technology for the benefit of all people.

IFIP is a non-profitmaking organization, run almost solely by 2500 volunteers. It operates through a number of technical committees, which organize events and publications. IFIP's events range from an international congress to local seminars, but the most important are:

- the IFIP World Computer Congress, held every second year;
- open conferences;
- working conferences.

The flagship event is the IFIP World Computer Congress, at which both invited and contributed papers are presented. Contributed papers are rigorously refereed and the rejection rate is high.

As with the Congress, participation in the open conferences is open to all and papers may be invited or submitted. Again, submitted papers are stringently refereed.

The working conferences are structured differently. They are usually run by a working group and attendance is small and by invitation only. Their purpose is to create an atmosphere conducive to innovation and development. Refereeing is less rigorous and papers are subjected to extensive group discussion.

Publications arising from IFIP events vary. The papers presented at the IFIP World Computer Congress and at open conferences are published as conference proceedings, while the results of the working conferences are often published as collections of selected and edited papers.

Any national society whose primary activity is in information may apply to become a full member of IFIP, although full membership is restricted to one society per country. Full members are entitled to vote at the annual General Assembly, National societies preferring a less committed involvement may apply for associate or corresponding membership. Associate members enjoy the same benefits as full members, but without voting rights. Corresponding members are not represented in IFIP bodies. Affiliated membership is open to non-national societies, and individual and honorary membership schemes are also offered.

# Systems Implementation 2000

**IFIP TC2 WG2.4 Working Conference on  
Systems Implementation 2000: Languages, methods and tools  
23–26 February 1998, Berlin, Germany**

Edited by

**R. N. Horspool**

*University of Victoria*

*Victoria*

*Canada*



**SPRINGER INTERNATIONAL PUBLISHING, CHAM**

First edition 1998

© 1998 IFIP International Federation for Information Processing  
Originally published by Chapman & Hall in 1998  
Softcover reprint of the hardcover 1st edition 1998

**I<sup>T</sup>P** Thomson Science is a division of International Thomson Publishing

ISBN 978-1-5041-2873-5      ISBN 978-0-387-35350-0 (eBook)  
DOI 10.1007/978-0-387-35350-0

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission in writing of the publishers. Applications for permission should be addressed to the rights manager at the London address of the publisher.

The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

A catalogue record for this book is available from the British Library

© Printed on acid-free text paper, manufactured in accordance with ANSI/NISO Z39.48-1992 (Permanence of Paper).

## CONTENTS

---

|                                                                                                                                                                       |            |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <b>Introduction</b>                                                                                                                                                   | <b>vii</b> |
| <b>Keynote Presentation I</b>                                                                                                                                         |            |
| 1 Evolutionary software engineering: a component-based approach<br><i>P. Klint and C. Verhoef</i>                                                                     | 1          |
| <b>Software Development</b>                                                                                                                                           |            |
| 2 Agendas – A concept to guide software development activities<br><i>M. Heisel</i>                                                                                    | 19         |
| 3 A new methodology for the development of CSCW systems<br><i>P. Ray</i>                                                                                              | 33         |
| 4 Managing requirements evolution and change<br><i>W. Lam</i>                                                                                                         | 45         |
| 5 The constraint logic paradigm: declarativity, efficiency, and flexibility<br><i>U. Geske, H-J. Goltz, U. John. and A. Wolf</i>                                      | 59         |
| <b>Java Evaluation</b>                                                                                                                                                |            |
| 6 Java as a systems programming language: three case studies<br><i>J.M. Bishop</i>                                                                                    | 72         |
| 7 Operating and window systems will never strike back or<br>independence day for Java developers<br><i>N. Fricke, C. Lilienthal, M. Lippert, S. Roock and H. Wolf</i> | 86         |
| 8 Java in high-performance networking applications<br><i>I. Azbel, A. Wynne, D. Cook and K. MacGregor</i>                                                             | 100        |
| <b>Software Architectures</b>                                                                                                                                         |            |
| 9. Representing structural requirements in software architecture<br><i>V. Ambriola and V. Gervasi</i>                                                                 | 114        |
| 10 An introduction to evolution of large systems based on<br>Software Architecture<br><i>J.C. Dueñas and G. León</i>                                                  | 128        |
| 11 Modeling and simulating optical computing architectures<br><i>I.R. Jones Jr. and V.P. Heuring</i>                                                                  | 140        |
| <b>Domain Specific Systems</b>                                                                                                                                        |            |
| 12 Compositional design and implementation of domain-specific languages<br><i>U. Kastens and P. Pfahler</i>                                                           | 152        |
| 13 A knowledge-based methodology applied to linguistic engineering<br><i>P. Martínez and A. García-Serrano</i>                                                        | 166        |

|                                |                                                                                                                          |     |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------|-----|
| 14                             | Attribute grammars in the functional style<br><i>S.D. Swierstra and P.R. Azero</i>                                       | 180 |
| <b>Keynote Presentation II</b> |                                                                                                                          |     |
| 15                             | Compiling at 1000 MHz and beyond<br><i>D.B. Wortman</i>                                                                  | 194 |
| <b>Software Components</b>     |                                                                                                                          |     |
| 16                             | On the architecture of software component systems<br><i>M. Franz</i>                                                     | 207 |
| 17                             | Meta-programming composers in second-generation component systems<br><i>U. Assmann</i>                                   | 221 |
| <b>Specifications</b>          |                                                                                                                          |     |
| 18                             | Object-oriented software reusability through formal specifications<br><i>L.M. Favre and G.M. Diez</i>                    | 235 |
| 19                             | Using many-sorted natural semantics to specify and generate semantic analysis<br><i>S. Glesner and W. Zimmermann</i>     | 249 |
| 20                             | TTT – A simple type-checked C language abstract data type generator<br><i>W.J. Toetenel</i>                              | 263 |
| <b>Concurrent Systems</b>      |                                                                                                                          |     |
| 21                             | LEAD: a linguistic approach to dynamic adaptability for practical applications<br><i>N. Amano and T. Watanabe</i>        | 277 |
| 22                             | A language adaptation architecture for reflective concurrent systems<br><i>C.H. Lin, E.H. Huang and T. Elrad</i>         | 291 |
| 23                             | A temporal logic language for system implementation<br><i>Z. Tang</i>                                                    | 305 |
| <b>WWW Technology</b>          |                                                                                                                          |     |
| 24                             | Evolution of WWW to support new integrated services<br><i>A. Dede, T. Zahariadis and G. Stassinopoulos</i>               | 319 |
| 25                             | Design and implementation of database powered web systems – experiences from the DEMETER project<br><i>K.M. Goeschka</i> | 333 |
| <b>Index of contributors</b>   |                                                                                                                          | 345 |
| <b>Keyword index</b>           |                                                                                                                          | 347 |

## INTRODUCTION

---

This book presents the proceedings of a conference with a very special flavour. Every year, hundreds of conferences are available to computer scientists, each focussing on a special area. *Systems Implementation 2000: Languages, Methods and Tools* is a one-off event with a very special mandate. It brings together scientists from all over the world to consider the issue central to the survival of computer systems: how should they best be programmed?

With so many conferences every year, a one-off event such as this one has to offer something different, and indeed it does. Breadth is achieved by looking at eight areas, from software components to Java performance, from concurrent systems to specifications; and depth by presenting full papers in the proceedings at the conference. Quality is achieved by a meticulous refereeing process, through IFIP Working Group 2.4, with fewer than half the submitted papers being accepted. Finally, a variety of viewpoints is evident from the fact that papers were submitted from 23 countries and accepted from 14 - Argentina, Australia, Austria, Canada, China, Germany, Greece, Italy, Japan, Spain, South Africa, The Netherlands, United Kingdom, United States of America. SI2000 is truly a world-wide conference, covering global issues of importance in the next century. We trust that those attending will take back to their countries revitalised hope for the future of software.

In putting together such a conference, the vision and hard work of many people are essential. I would like to thank in particular the secretary of WG2.4, Nigel Horspool, who has edited these proceedings, maintained the conference website and been a source of good counsel throughout. Thanks to all the members and observers of WG2.4 who assisted with the refereeing process, and also to Adam Czajka, Bruce Kapron, Derrick Kourie, Jacek Martinek, Stanislaw Szejko and Michael Zastre. The local organisation has been ably run by Stefan Jaehnichen and Nikola Serbedzija. The accompanying events, including a tour of the world-famous Pergamon Museum of Antiquities in Berlin, are due to their imagination and energy.

Finally, we gratefully acknowledge the professional assistance received from the editorial and production departments of Chapman and Hall, especially Alison Provan. Their willingness to adapt to our tight schedule has meant that the proceedings could be available at the conference. We trust that this book will be as useful and interesting to the attendees as it will be to those who buy it later.

Judith M Bishop  
*Chairman, IFIP WG2.4 (Systems Implementation Languages)*