

Edited by G. Goos and J. Hartmanis

Advisory Board: W. Brauer D. Gries J. Stoer



A. Endres H. Weber (Eds.)

# Software Development Environments and CASE Technology

European Symposium  
Königswinter, June 17-19, 1991  
Proceedings

**Springer-Verlag**

Berlin Heidelberg New York  
London Paris Tokyo  
Hong Kong Barcelona  
Budapest

## Series Editors

Gerhard Goos  
GMD Forschungsstelle  
Universität Karlsruhe  
Vincenz-Priessnitz-Straße 1  
W-7500 Karlsruhe, FRG

Juris Hartmanis  
Department of Computer Science  
Cornell University  
Upson Hall  
Ithaca, NY 14853, USA

## Volume Editors

Albert Endres  
IBM Laboratories  
Schönaicher Str. 220, W-7030 Böblingen, FRG

Herbert Weber  
Universität Dortmund, Fachbereich Informatik  
Postfach 50 05 00, W-4600 Dortmund, FRG

CR Subject Classification (1991): D.2

ISBN 3-540-54194-2 Springer-Verlag Berlin Heidelberg New York  
ISBN 0-387-54194-2 Springer-Verlag New York Berlin Heidelberg

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 other ways, and storage in data banks. Duplication of this publication or parts thereof is only permitted under the provisions of the German Copyright Law of September 9, 1965, in its current version, and a copyright fee must always be paid. Violations fall under the prosecution act of the German Copyright Law.

© Springer-Verlag Berlin Heidelberg 1991  
Printed in Germany

Printing and binding: Druckhaus Beltz, Hemsbach/Bergstr.  
2145/3140-543210 - Printed on acid-free paper

## Preface

The term *Software Development Environment* has been introduced in the 1980's to designate integrated sets of tools, techniques and processes that assist in the systematic development of software products. Software development environments are intended to support all phases of the software development cycle, namely requirements definition, design, implementation, test, and maintenance. Many environments provide explicit support for the different roles in a project, such as development, management and quality assurance.

While tools that support the implementation and test phases have quite a long tradition, powerful tools that also support the requirements definition and the design process have appeared only recently. Since these tools allow the employment of well-known software engineering methods, the term *Computer Aided Software Engineering* (CASE) has been coined. CASE tools typically make use of graphic manipulation facilities and rely on some form of development database to exchange various types of design objects between tools.

Facilitated by the availability of powerful workstations, implementations of these technologies are now within reach of every software developer. Their proper use and thoughtful introduction can take an organization a significant step forward towards higher development productivity and improved product quality. In spite of this, we still observe a rather slow acceptance of these concepts in the industrial practice. This seems to be caused by a number of reasons:

- Many of the tools currently offered in the market place serve as stand-alone tools only and do not allow a true integration into development environments, nor do they support large team efforts.
- Sometimes, potential users of software engineering methods and software development environments are expecting a kind of magic or instantaneous solution to all their problems, the famous "silver bullet" (as Fred Brooks called it), and do not plan for the adjustment of available concepts to their environment.
- Other users fail because they underestimate the prerequisites on their part and the cost incurred for deploying software engineering methods and software development environments throughout their organization.

For this symposium a special attempt was made to bring together application development managers and top specialists involved in the selection and introduction of software development

tools with leading developers of such tools and well-known authorities from the research community. The proceedings therefore contain first-hand information on practical experiences and requirements, development directions and strategies, and key research issues and results. All speakers were invited based on recommendations of an international programme committee. Although we seem to have achieved an almost balanced situation as far as the number of papers in each category is concerned, we did not expect to reach uniformity in comprehensiveness and style. If more time and more guidance were given to the authors, maybe more could have been done in this respect. But diversity exists, and cannot be ignored.

The symposium was organized by the special interest group (Fachgruppe) Software Engineering of the Gesellschaft für Informatik e.V.(GI), Bonn, the German informatics society, in cooperation with the Gesellschaft für Mathematik und Datenverarbeitung mbH (GMD), Birlinghoven, the leading German research institution in computer science, and the Steering Committee for the European Software Engineering Conferences (ESEC), a coordinating body sponsored by several European professional societies (such as AFCET, AICA, BCS, GI, ÖGI and SI) to foster joint activities in the field of software engineering.

We would like to acknowledge the help and advice of the members of the Organizing Committee and the cooperation and dedication of the authors and the other contributors and supporters. We hope that participants and readers will benefit from these efforts by putting them in a better position to make well-founded decisions in a critical area of today's information processing business.

Albert Endres, Symposium Chairman  
Herbert Weber, Program Chairman

# Contents

## Development Directions and Strategies

Systems Engineering Environments of ATMOSPHERE <i>Henk Obbink</i>	1
Building IPSE's by Combining Heterogeneous CASE Tools <i>Peter Göhner</i>	18
On the Functional and Architectural Integration of CASE Systems <i>Günter Merbeth</i>	31
Development of a Method Driven CAS <sup>2</sup> E Tool <i>Jonah Z. Lavi, Michael Winokur</i>	44
VSF and its Relationship to Open Systems and Standard Repositories <i>John N. Pocock</i>	53
Adding Control Integration to PCTE <i>Huw Oliver</i>	69
Meta-CASE Technology <i>Albert Alderson</i>	81

## Application Experience and Requirements

Management Issues in Software Development <i>Rainer Burchett</i>	92
Standardization of Software Development Environments <i>Adolf Peter Bröhl</i>	99
A Software Development Environment for CIM Applications <i>Lothar Köhler</i>	109
A Repository and Other Tools in a Commercial Development Center <i>Dieter Steinbauer</i>	114
Lessons Learned from DOMINO and GRAPES <i>Johann Wagner</i>	130
Should CASE be Application Dependent? <i>Peter Hruschka</i>	143

## Research Issues and Results

Utilizing Fifth Generation Technology in Software Development Tools <i>Wolfgang Henhapl, Stefan Kaes, Gregor Snelting</i>	153
Integrated Software Components: A Paradigm for Control Integration <i>Dominique Clément, Vincent Prunet, Francis Montagnac</i>	167
Formal Methods in Software Development: Requirements for a CASE <i>Dines Bjørner, Søren Prehn</i>	178
Modeling of Software Architectures: Importance, Notions, Experiences <i>Manfred Nagl</i>	211
A Configurable Framework for Method and Tool Integration <i>Jeff Kramer, Anthony Finkelstein</i>	233
Recent Findings in Software Process Maturity <i>Watts S. Humphrey</i>	258
Validation and Verification of Software Process Models <i>Volker Gruhn</i>	271