

Lecture Notes in Computer Science

2269

Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Springer

Berlin

Heidelberg

New York

Barcelona

Hong Kong

London

Milan

Paris

Tokyo

Stephan Diehl (Ed.)

Software Visualization

International Seminar
Dagstuhl Castle, Germany, May 20-25, 2001
Revised Papers



Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editor

Stephan Diehl
Saarland University, FR Informatik
PO Box 15 11 50, 66041 Saarbrücken, Germany
E-mail: diehl@acm.org

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Software visualization : international seminar, Dagstuhl castle, Germany,
May 20 - 25, 2001 ; revised papers / Stephan Diehl (ed.). - Berlin ;
Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ;
Tokyo : Springer, 2002
(Lecture notes in computer science ; Vol. 2269)
ISBN 3-540-43323-6

CR Subject Classification (1998):D.2, D.1.7, F.2, G.2, H.2, I.3

ISSN 0302-9743

ISBN 3-540-43323-6 Springer-Verlag 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-Verlag. Violations are liable for prosecution under the German Copyright Law.

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

<http://www.springer.de>

© Springer-Verlag Berlin Heidelberg 2002
Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Stefan Sossna
Printed on acid-free paper SPIN 10846115 06/3142 5 4 3 2 1 0

Preface

Researchers in software visualization develop and investigate methods and use of computer-graphical representations of different aspects of software such as its static structure, its concrete or abstract execution as well as its design and evolution.

Since Goldstein's and von Neumann's demonstration of the usefulness of flowcharts in 1947 visual representations have played an important role in understanding and designing programs. Software visualization is done in many areas of computer science, but often not realized as a field of its own. As a result papers are published at conferences and workshops of these areas reinventing the wheel over and over again.

The planning for this book started at the Dagstuhl Seminar on Software Visualization during May 2001. The goal of this seminar was to bring together practitioners and researchers working in the area of software visualization as well as those working in related areas including database visualization, graph drawing, and visual programming. Discussions and presentations at the seminar were not restricted to theoretical foundations and technical applications. They also addressed psychological and educational aspects.

The intent of this book is to present the state of the art in software visualization. To this aim it contains introductory papers and original work. More than 60 authors contributed to this volume. It is divided into five chapters:

- algorithm animation,
- software visualization and software engineering,
- software visualization and education,
- graphs in software visualization,
- and perspectives of software visualization.

Each chapter starts with an introduction surveying previous and current work and providing extensive bibliographies.

Eventually we hope that this volume will foster software visualization and its impact on the way we teach, learn, and design programs.

The editor would like to gratefully acknowledge the support provided by the authors, Springer-Verlag, and the Dagstuhl staff during the seminar and the making of this volume.

Foreword by the Organizers

The International Dagstuhl Seminar on Software Visualization was held in May 2001 at the International Research and Conference Center for Computer Science in Schloss Dagstuhl, Germany. Dagstuhl seminars are one-week meetings which bring together the most significant researchers on important topics in computer science. Participation is by invitation only.

It is often said that humans have never before created any artifacts which are as complex as today's software systems. As a result creating, maintaining, understanding, and teaching software is a challenging task. Software is neither matter nor energy, it is just a kind of information. Sometimes the representation and the information itself are confused. Software visualization is concerned with visually representing different aspects of software including its structure, execution, and evolution. So far, research on software visualization has mostly been motivated by its potential to support teaching. Many systems have been developed to facilitate the production of algorithm animations. At Dagstuhl software engineers and re-engineers have repeatedly argued that there is a strong need for software visualization in their areas. Here further research includes the use of techniques from information visualization to display software metrics, graph layout and graph animations to show the structure and changes in software systems, and program animation for debugging. At the seminar more than 50 researchers from all around the world discussed the state of the art as well as challenging questions for the future of software visualization. The program included 38 presentations and 15 system demonstrations, as well as several sessions for group discussions. Participants of the seminar volunteered

- to compile a post seminar proceedings, which is to be published as a Springer LNCS state-of-the-art survey,
- to create a repository with algorithm animations, and software visualization tools
- to initiate an international conference series on software visualization.

We feel that the seminar was a seminal event. The future will tell whether it reached its ambitious goals to form a community and raise awareness of software visualization as a challenging and important research field of its own.

December 2001

Stephan Diehl
Peter Eades
John Stasko

Table of Contents

1. Algorithm Animation

<i>with an Introduction by Andreas Kerren and John T. Stasko</i>	1
Specifying Algorithm Visualizations: Interesting Events or State Mapping? ..	16
<i>Camil Demetrescu, Irene Finocchi, and John T. Stasko</i>	
Perspectives on Program Animation with Jeliot	31
<i>Mordechai Ben-Ari, Niko Myller, Erkki Sutinen, and Jorma Tarhio</i>	
Animating Algorithms Live and Post Mortem	46
<i>Stephan Diehl, Carsten Görg, and Andreas Kerren</i>	
Visualising Objects: Abstraction, Encapsulation, Aliasing, and Ownership ..	58
<i>James Noble</i>	
Algorithm Animation Using Data Flow Tracing	73
<i>Jaroslav Francik</i>	
GeoWin: A Generic Tool for Interactive Visualization of Geometric Algorithms	88
<i>Matthias Bäsken and Stefan Näher</i>	
Algorithm Animation Systems for Constrained Domains	101
<i>Ayellet Tal</i>	
Algorithm Animation for Teaching	113
<i>Rudolf Fleischer and Luděk Kučera</i>	

2. Software Engineering

<i>with an Introduction by Rym Mili and Renee Steiner</i>	129
Software Visualization for Reverse Engineering	138
<i>Rainer Koschke</i>	
Visualizing the Execution of Java Programs	151
<i>Wim De Pauw, Erik Jensen, Nick Mitchell, Gary Sevitsky, John Vlissides, and Jeaha Yang</i>	
JaVis: A UML-Based Visualization and Debugging Environment for Concurrent Java Programs	163
<i>Katharina Mehner</i>	
JAVAVIS: Automatic Program Visualization with Object and Sequence Diagrams Using the Java Debug Interface (JDI)	176
<i>Rainer Oechsle and Thomas Schmitt</i>	
Visualizing Memory Graphs	191
<i>Thomas Zimmermann and Andreas Zeller</i>	

3. Software Visualization and Education

<i>with an Introduction by John Domingue</i>	205
Structure and Constraints in Interactive Exploratory Algorithm Learning .	213
<i>Nils Faltin</i>	
A Language and System for Constructing and Presenting Low Fidelity Algorithm Visualizations	227
<i>Christopher Hundhausen and Sarah Douglas</i>	
Towards a Taxonomy of Network Protocol Visualization Tools	241
<i>Pilu Crescenzi and Gaia Innocenti</i>	
Understanding Algorithms by Means of Visualized Path Testing	256
<i>Ari Korhonen, Erkki Sutinen, and Jorma Tarhio</i>	
Hypertextbooks: Animated, Active Learning, Comprehensive Teaching and Learning Resources for the Web	269
<i>Rockford J. Ross and Michael T. Grinder</i>	

4. Graphs in Software Visualization

<i>with an Introduction by Petra Mutzel and Peter Eades</i>	285
On the Visualization of Java Programs	295
<i>Holger Eichelberger and J. Wolff von Gudenberg</i>	
Graph Drawing Algorithm Engineering with AGD	307
<i>Carsten Gutwenger, Michael Jünger, Gunnar W. Klau, Sebastian Leipert, and Petra Mutzel</i>	
An Overview of the GXL Graph Exchange Language	324
<i>Andreas Winter, Bernt Kullbach, and Volker Riediger</i>	
Call Graph and Control Flow Graph Visualization for Developers of Embedded Applications.....	337
<i>Alexander A. Evstiougov-Babaev</i>	

5. Future Perspectives

<i>with an Introduction by Stephan Diehl</i>	347
Visualization for the Mind's Eye	354
<i>Nelson Baloian and Wolfram Luther</i>	
The <i>rube</i> Framework for Personalized 3-D Software Visualization	368
<i>John F. Hopkins and Paul A. Fishwick</i>	
Algorithm Explanation: Visualizing Abstract States and Invariants.....	381
<i>Reinhard Wilhelm, Tomasz Müldner, and Raimund Seidel</i>	
Visualisation and Debugging of Decentralised Information Ecosystems	395
<i>Rolf Hendrik van Lengen and Jan-Thies Bähr</i>	

Author Index	405
---------------------------	-----