Software Visualization

Stephan Diehl

Software Visualization

Visualizing the Structure, Behaviour, and Evolution of Software

With 124 Figures, including 75 in Colour, and 5 Tables

Springer

Author

Stephan Diehl

Universität Trier
Fachbereich Informatik
54286 Trier, Germany
diehl@uni-trier.de

Library of Congress Control Number: 2007923067

ACM Computing Classification (1998): D.2, I.3.8, J.6

ISBN 978-3-540-46504-1 Springer Berlin Heidelberg New York

Typeset by the author
Production: LE-TEX Jelonek, Schmidt & Vöckler GbR, Leipzig
Cover design: KünkelLopka Werbeagentur, Heidelberg

Printed on acid-free paper 45/3100/YL - 5 4 3 2 1 0

To Christine, Luca, and Jean-Luc

Preface

Software systems are designed, implemented, tested, debugged, analyzed, and maintained by many changing developers. All these tasks can be facilitated by visualization.

In this book we give an overview of the various areas of software visualization, the art and science of generating visual representations of various aspects of software and its development process.

In contrast to visual programming and diagramming for software design, software visualization is not so much concerned with the construction, but with the analysis of programs and their development process.

So far, there exist only anthologies and proceedings about software visualization. This book is the first textbook on software visualization. Although written mostly for graduate students, the book is also a valuable resource for researchers as it provides a broad and systematic overview of the area with many pointers to literature and systems for further study.

As the field of software visualization is growing fast, the book is not meant to be comprehensive, but we have attempted to select seminal work as well as promising new approaches to illustrate some emerging principles in the field. Each chapter is followed by a list of exercises including both pen&paper exercises, as well as programming tasks.

This book is aimed at graduate students and researchers who are new to the field of software visualization. The book is meant to be read from end to end, though some readers may want to skip some of the more formal sections. Ideally, after reading the book, the reader will be able to

- identify recurring concepts in various areas of software visualization;
- understand the purpose of various visualization techniques;
- appreciate the use of visualization in software engineering.

We assume that the reader will have some programming experience, preferably in Java, and some basic knowledge of software engineering terminology. No prior knowledge of software visualization is required.

Additional material related to this book, including examples and program code for exercises, can be found at the following Web address:

`http://www.eposoft.org/svbook`

March 2007 *Stephan Diehl*

Contents