Foundations of Component-Based Systems

This collection of articles by well-known experts is intended for researchers in computer science, practitioners of formal methods, and computer programmers working in safety-critical applications or in the technology of component-based systems. The work brings together, for the first time, several elements of this area that are fast becoming the focus of much current research and practice in computing.

The introduction by Clemens Szyperski gives a snapshot of the current state of the field. About half the articles deal with theoretical frameworks, models, and systems of notation; the rest of the book concentrates on case studies by researchers who have built prototype systems and present findings on architecture verification. The emphasis is on advances in the ideas behind component-based systems; how to design and specify reusable components; and how to reason about, verify, and validate systems from components.

Gary T. Leavens is Associate Professor in the Department of Computer Science at Iowa State University. He received his Ph.D. from MIT in 1989 and has taught at Iowa State University ever since. Dr. Leavens has written for such journals as *ACM TOSEN*, *ACM TOPLAS*, *Theoretical Computer Science*, *Acta Informatica*, and *Theory and Practice of Object Systems*. He serves on the program committees of well-known conferences such as OOPSLA and ICSE. His research has been funded by the U.S. National Science Foundation.

Murali Sitaraman is Associate Professor in the Department of Computer Science and Electrical Engineering at West Virginia University. He received his Ph.D. at Ohio State University and has taught at West Virginia University since 1990. Dr. Sitaraman has written for such journals as *IEEE Transactions on Software Engineering*, *Software Practice and Experience*, *Formal Aspects of Computing*, and *IEEE Software*. He was program chair of the IEEE Computer Society International Conference on Software Reuse in 1996 and has served on program committees for several major conferences.

Foundations of Component-Based Systems

Edited by

GARY T. LEAVENS
Iowa State University

MURALI SITARAMAN
West Virginia University

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE UNIVERSITY PRESS
Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore,
São Paulo, Delhi, Dubai, Tokyo, Mexico City

Cambridge University Press
The Edinburgh Building, Cambridge CB2 8RU, UK

Published in the United States of America by Cambridge University Press, New York

www.cambridge.org
Information on this title: www.cambridge.org/9780521155694

First published 2000
First paperback edition 2010

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

Library of Congress Cataloguing in Publication Data

Foundations of component-based systems 1 edited by Gary T. Leavens, Murali Sitaraman.
p.cm.
ISBN 0-521-77164-1
1. Component software. 2. System design. I. Leavens, Gary T. II. Sitaraman, Murali.

QA76.76.C66 F68 2000
005.1-dc21 99-049686

ISBN 978-0-521-77164-1 Hardback
ISBN 978-0-521-15569-4 Paperback

Contents

Preface

Component-based software construction has become a central focus of software engineering research and computing practice. There is a near-universal recognition in the field that development of high-quality systems on time is possible only through assembly of well-conceived and prefabricated software components.

This volume brings together, for the first time, programming and specification issues as well as framework, architecture, and distributed computing issues that should be considered in designing component-based systems. It simultaneously tries to lay a foundation to bridge a spectrum of approaches that span current component-based technology and formal foundational research. The technological chapters focus on component structure and integration issues, providing a basis for the latter chapters that focus on component semantics in sequential and concurrent setting. Taken together, these chapters should benefit software engineering practitioners in enhancing component-based construction practice and researchers in establishing a connection to practical technology. They should also provide new research directions to computer science and engineering graduate students for advancing the field.

The volume begins with a chapter by Szyperski, which summarizes current technologies, including COM, CORBA, and JavaBeans. This chapter highlights the importance and role of component-based construction for modern computing. It also outlines essential problems to be solved for component-based software to become both reliable and effective.

Following this initial chapter, Part One of the volume focuses on elements of frameworks and architectures for component-based composition. Though the chapters in this section use particular languages and notations to illustrate the ideas, the central themes of the chapters are generally applicable to all component-based software construction.

In Part One, the chapter by Luckham, Vera, and Meldal explains that an architecture is a specification of the components and communication among them, and elaborates on the concepts needed to support this view. Garlan, Monroe, and Wile introduce ACME as a common representation for software architectures and as an enabler to integrate systems built using alternative architecture definition languages.

The chapter by Lumpe, Achermann, and Nierstrasz defines the requirements of a flexible software composition language and provides formal semantic foundations to facilitate precise specification and formal reasoning. Chen and Cheng describe criteria for matching specifications and explain what constitutes a reuse-ensuring match.

Part Two deals with aspects of formal specification and verification, with emphasis on object-based software construction. The chapters discuss behavioral subtyping, specification and verification, and preservation of behavior when objects of one type are converted to another. Leavens and Dhara give a background and survey of behavioral subtyping, which is a relationship between types that allows modular specification and verification of object-oriented software. Müller and Poetzsch-Heffter explain a modular technique for specifying and verifying object-oriented components. Wing and Ockerbloom's chapter focuses on guaranteeing consistent observable behavior when converting objects of one type to another.

Part Three concentrates on formal models and formal semantics of components and compositions. The chapters in this part describe complementary approaches for understanding components and compositions.

In Part Three, the chapter by Bergner, Rausch, Sihling, Vilbig, and Broy gives a formal model that encompasses both components and object-oriented features of programming languages. They use this model to describe the meaning of commonly used graphical description techniques (such as some diagram forms in the UML). Gibson, Weide, Pike, and Edwards provide a formal model of parameterized component-based (software) systems that facilitates modular reasoning about collections of interacting components. Goguen and Tracz describe an algebraic approach to software engineering. Their chapter introduces module expressions to compose components and provides an implementation-oriented semantics for composition.

Part Four of the volume contains chapters on reactive and distributed computing, two critical aspects of component-based systems, and modern software practice. The chapter by Lano, Bicarregui, Maibaum, and Fiadeiro describes a modular, declarative approach for specification of reactive systems. The approach is suitable for model-based design notations such as VDM and B. Garland and Lynch's chapter presents a new language for structured modeling of distributed computing systems, using a mathematical I/O automaton model as the basis. The chapter also provides an overview of design and analysis tools that can be developed using the model.

Though we have organized this volume beginning with issues on languages and frameworks, and proceeding to techniques for specification, verification, formal models, and distribution, the individual parts and chapters in this volume are self-contained. A reader or teacher should be able to choose the order in which to read or discuss the chapters.

The chapters in this volume have not been previously published. They were particularly solicited for this volume from experts in the field. To ensure high quality, all chapters were peer reviewed. Every chapter, except for one, had at least two

reviewers. In every case, the reviewers provided detailed and timely feedback to the authors for revision. Our sincere thanks are due to the reviewers, including: Franz Achermann, Uwe Assmann, Gerald Baumgartner, Manfred Broy, Jack Callahan, Betty Cheng, Paolo Ciancarini, James Donahue, Stephen Edwards, Bernd Fischer, Rustan Leino, Ali Mili, Anna Mikhajlova, Oscar Nierstrasz, John Penix, Johannes Sametinger, Oleg Sheyner, Judith Stafford, and Raymie Stata.

Our sincere thanks to Lauren Cowles at Cambridge University Press for her support and advice during this project, and to Ernie Haim for his careful eye and help in the production of this book. We thank Addison-Wesley for giving permission to Szyperski to derive his chapter from his book *Component Software: Beyond Object-Oriented Programming* (Addison-Wesley, 1998). We also thank the U.S. National Science Foundation, the U.S. Defense Advanced Research Projects Agency, and our institutions for supporting this editorial work. Thanks to Janet from Gary and to Susan and Nathan from Murali for their love and support.

Gary T. Leavens, Ames, Iowa
Murali Sitaraman, Morgantown, West Virginia
January 14, 2000