COMPONENT-BASED SOFTWARE DEVELOPMENT: CASE STUDIES

SERIES ON COMPONENT-BASED SOFTWARE DEVELOPMENT

Aims and Scope

Component-based software development (CBD) is concerned with software development by using independently produced components, e.g. commercial off-the-shelf (COTS) software. It is an emerging discipline that promises to take software engineering into a new era. Building on the achievements of object-oriented software construction, CBD aims to deliver software engineering from a “cottage industry” into an “industrial age for Information Technology,” whereby software can be assembled in the manner that hardware systems are currently constructed from kits of parts. Although the idea of using components to build complex systems is as old as the software engineering discipline itself, it is only recently that CBD has become the focus of research for a sizeable international community.

This series aims to serve this community by publishing all kinds of material on CBD: state-of-the-art reports and surveys, conference proceedings, collected works, research monographs, as well as textbooks. It should therefore provide a key forum for CBD researchers, practitioners and students worldwide.

Call for Contributions

If you have suitable ideas or material for the series, please contact the Editor-in-Chief:

Kung-Kiu Lau
Department of Computer Science
The University of Manchester
Manchester M13 9PL
United Kingdom
kung-kiu@cs.man.ac.uk

COMPONENT-BASED SOFTWARE DEVELOPMENT: CASE STUDIES

Kung-Kiu Lau

The University of Manchester, UK

editor

Published by

World Scientific Publishing Co. Pte. Ltd.

5 Toh Tuck Link, Singapore 596224

USA office: Suite 202, 1060 Main Street, River Edge, NJ 07661

UK office: 57 Shelton Street, Covent Garden, London WC2H 9HE

British Library Cataloguing-in-Publication Data

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

COMPONENT-BASED SOFTWARE DEVELOPMENT

Case Studies

Copyright © 2004 by World Scientific Publishing Co. Pte. Ltd.

All rights reserved. This book, or parts thereof, may not be reproduced in any form or by any means, electronic or mechanical, including photocopying, recording or any information storage and retrieval system now known or to be invented, without written permission from the Publisher.

For photocopying of material in this volume, please pay a copying fee through the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, USA. In this case permission to photocopy is not required from the publisher.

ISBN 981-238-828-1

Printed in Singapore.

Preface

Component-based Software Development (CBD) is an emerging discipline that promises to take Software Engineering into a new era. Building on the achievements of Object-Oriented Software Construction, CBD aims to deliver Software Engineering from a ‘cottage industry’ into an ‘industrial age for Information Technology’, whereby software can be assembled from components, in the manner that hardware systems are constructed from kits of parts nowadays.

As with any emerging discipline, there is as yet no standard terminology in CBD. No universal agreement has been reached on the very definition of what a software component is, and there is even an alternative to the acronym CBD itself, namely CBSE (Component-based Software Engineering), which is just as widely used. Even less clear is how the whole field will develop in the near future, let alone the longer term. Therefore it would seem more sensible (and feasible) to survey the current state of CBD, as reflected by activities that have been going on recently under the banner of CBD, than to predict future trends. The primary aim of this volume therefore is to attempt to provide such a snapshot. It is hoped that the survey would have the desirable side-effect of indicating pointers to future trends.

The articles in this volume report on case studies into a sample of the myriad aspects of CBD. Case studies are loosely defined as self-contained fixed-term investigations with a finite set of clearly defined objectives and measurable outcomes. In a new field, where mature theories do not yet exist, case studies provide an informative glimpse into the current concerns and issues.

The first part of the book deals with COTS (commercial off-the-shelf) components. These are the first commercially available software components. Typically they are executable binaries that come with no proper specifications, no source code and no guarantees. Chapter 1 is a survey of approaches to constructing software using COTS components. Chapter 2 describes one approach to specifying COTS components, and Chapter 3 reports on a case study on an acquisition process for third-party COTS components.

Various methodologies for CBD have been proposed. Chapter 4 describes a case study on using the KobrA method, and Chapter 5 argues for an aspect-oriented approach to the design and implementation stages of CBD life cycle.

A crucial property of CBD software is compositionality, i.e. how to calculate or predict

properties of a composite from those of its constituents. Indeed, compositional or predictable assembly is a pre-requisite for CBD to succeed in general. Chapter 6 presents a theory of properties of component-based systems. Chapter 7 shows an approach to measuring properties of embedded control systems.

Software testing is also important for component software. Chapter 8 describes a case study on testing CORBA software.

What about the role of formality in CBD? Chapter 9 describes a formal approach to using components in the requirements capture stage of the software life cycle, and Chapter 10 presents a formal approach to specifying and adapting components.

Reflecting the diverse nature of CBD, the last part of the book deals with components used in Grid Computing. Chapter 11 describes a case study on a CORBA-based architecture for legacy applications in Grid Computing, and Chapter 12 reports on experiments in Problem Solving Environments for numerical computations.

Finally, as editor, I would like to thank all the contributing authors. Personally I have enjoyed learning about their work. I wish them success in their CBD research, and I look forward to their future contributions. I would also like to thank the people who provided invaluable help with the reviews: Manuel Carro, Len Freeman, Bernd Fischer, Shui-Ming Ho, Tuomas Ihme, Gerald Kotonya, Rikard Land, Chang Liu, Zhiming Liu, Frank Lüders, John Penix, Judith Stafford and Anders Wall. To all the authors and the reviewers, I owe a debt of gratitude.

Manchester
September 2003

Kung-Kiu Lau

Contents

<i>Preface</i>	v
1. <i>A Survey of Proposals for Architecting Component Software</i>	1
T. Ihm (VTT Electronics, Finland)	
2. <i>Describing Specifications and Architectural Requirements of COTS Components</i>	35
L. Iribarne (University of Almería, Spain), J. M. Troya, A. Vallecillo (University of Málaga, Spain)	
3. <i>Definition of COTS Software Component Acquisition Process — The Case of a Telecommunication Company</i>	57
P. Ulkuniemi, V. Seppänen (University of Oulu, Finland)	
4. <i>The Library Systems Product Line: A Case Study Demonstrating the Kobra Method</i>	81
J. Bayer, D. Muthig (Fraunhofer Institute for Experimental Software Engineering, Germany)	
5. <i>Component-based System Design and Composition: An Aspect-oriented Approach</i>	109
P.J. Clemente, J. Hernández, J.M. Murillo, M.A. Pérez, F. Sánchez (University of Extremadura, Spain)	
6. <i>Properties of Software Systems Synthesized from Components</i>	129
D. Hamlet (Portland State University, USA), D. Mason, D. Woit (Ryerson Polytechnic University, Canada)	
7. <i>Component-based Measurement and Control Software for Embedded Systems</i>	159
W. Maydl, B. Sick, W. Grass (University of Passau, Germany)	

8.	<i>Fault-based Testing of CORBA Component Software</i>	187
	S. Ghosh (Colorado State University, USA), A.P. Mathur (Purdue University, USA)	
9.	<i>ARIFS Methodology: A Case Study</i>	211
	R.P. Díaz Redondo, J.J. Pazos Arias, A. Fernández Vilas (University of Vigo, Spain)	
10.	<i>REBOUND: A Framework for Automated Component Adaptation</i>	235
	J. Penix (NASA Ames, USA), P. Alexander (University of Kansas, USA) .	
11.	<i>A Web-enabled Component-based Architecture for Legacy Applications</i>	261
	G. Wang, R. Robinson, S. Sripada (The Boeing Company, USA)	
12.	<i>Component-based Problem Solving Environments for Computational Science</i>	281
	M. Li (Brunel University, UK), O.F. Rana, D.W. Walker, M. Shields, Y. Huang (Cardiff University, UK)	
	<i>Index</i>	301