FORMAL METHODS FOR DISTRIBUTED PROCESSING

This book presents the current state of the art in the application of formal methods to object-based distributed systems. A major theme of the book is how to formally handle the new requirements arising from OO distributed systems, such as dynamic reconfiguration, encapsulation, subtyping, inheritance and real-time aspects. These may be supported either by enhancing existing notations, such as UML, LOTOS, SDL and Z, or by defining new notations, such as Actors, Pi-calculus and Ambients. The major specification notations and modelling techniques are introduced and compared by leading researchers, in several cases the inventors of the notations. The book also includes a description of approaches to the specification of nonfunctional requirements, which are typically needed in the specification of multimedia systems, and a discussion of security issues.

Researchers and practitioners in software design, object-oriented computing, distributed systems and telecommunications systems will gain an appreciation of the relationships between the major areas of concern and learn how the use of object-oriented-based formal methods provides workable solutions.

Howard Bowman is a senior lecturer in the Computing Laboratory at the University of Kent at Canterbury. He received his PhD from the University of Lancaster, where he was also a postdoctoral fellow. He is co-author of a book on specifying distributed multimedia systems using real-time temporal logics. His current research interests include the application of formal description techniques in object-oriented distributed systems and the formal specification and validation of multimedia systems (using real-time temporal logics).

John Derrick is a reader in the Computing Laboratory at the University of Kent at Canterbury. He received his DPhil from Oxford University and worked briefly at University College North Wales before joining STC Technology Ltd (now Nortel). In 1990 he joined the Computing Laboratory at the University of Kent. His interests include specification techniques for distributed systems, refinement and testing.

FORMAL METHODS FOR DISTRIBUTED PROCESSING

A Survey of Object-Oriented Approaches

Edited by

HOWARD BOWMAN
University of Kent at Canterbury

JOHN DERRICK
University of Kent at Canterbury

CAMBRIDGE UNIVERSITY PRESS

PUBLISHED BY THE PRESS SYNDICATE OF THE UNIVERSITY OF CAMBRIDGE
The Pitt Building, Trumpington Street, Cambridge, United Kingdom

CAMBRIDGE UNIVERSITY PRESS
The Edinburgh Building, Cambridge CB2 2RU, UK
40 West 20th Street, New York, NY 10011-4211, USA
10 Stamford Road, Oakleigh, VIC 3166, Australia
Ruiz de Alarcón 13, 28014 Madrid, Spain
Dock House, The Waterfront, Cape Town 8001, South Africa

http://www.cambridge.org

First published 2001

Printed in the United States of America

Typeface Computer Modern 10/12 pt. *System* $\LaTeX$ [AU]

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

Library of Congress Cataloging in Publication data

Formal methods for distributed processing : a survey of object-oriented approaches /
edited by Howard Bowman, John Derrick.
p. cm.
ISBN 0-521-77184-6
1. Electronic data processing – Distributed processing. 2. Object-oriented programming
(Computer science) I. Bowman, Howard, 1966– II. Derrick, John, 1963–

QA76.9.D5 F662 2001
005.1'17–dc21 2001025501

ISBN 0 521 77184 6 hardback

Contents

Preface

Overview

The aim of this book is to review and reflect on the major recent research developments in the application of formal methods to distributed processing. In doing so the book aims to provide:

- an introduction to modern object-oriented distributed systems;
- an introduction and comparison of the major specification notations and modelling techniques;
- an introduction to the specification of systems involving dynamic reconfiguration;
- a discussion of the role and use of subtyping;
- an introduction to concurrent object-oriented languages and reflection;
- a description of approaches to the specification of nonfunctional requirements, which are needed typically in the specification of multimedia systems;
- a discussion of the role of development architectures in distributed systems modelling.

A strength of the book is that it encapsulates a number of issues in a single text, and this is a feature that we believe readers will find valuable. After providing a background in modern object-oriented distributed systems, a number of themes are developed. Each theme provides a reflective survey of the state of the art of research written by some of the most prominent researchers in the field. We hope that this will enable the reader to gain an appreciation of the relationships between the major areas of concern and how the use of object-based formal methods seeks to provide workable solutions that address these concerns.

Specifying Distributed Systems

This book is about distributed systems, their inherent complexity and how we can specify and analyse such complex systems. In particular, we focus on how to apply formal methods to the design of open object-oriented distributed systems.

Distributed systems have become indispensable in modern society, and the growth

of the Internet and the World Wide Web have radically altered how we do business and disseminate information. This has been possible through the realisation of new techniques and technologies, and, in particular, the use of object orientation as an encapsulation mechanism has been vital in building component-based distributed system architectures.

One such architecture is that provided by the Open Distributed Processing (ODP) standard. This has defined a reference model that provides a framework within which distributed applications can be constructed. The CORBA middleware platform can be viewed as an example of realising and using the ODP architecture. In the first part of this book we provide an introduction to some of the important current issues in distributed systems, focusing in Chapter 2 on the ODP model. Although ODP is not the only relevant framework, it does offer a vehicle by which to discuss some of the major issues in distributed systems. In particular, the broad scope of ODP ensures that it provides a framework in which the spectrum of topics and techniques can be studied. These include such topics as multimedia, openness, dynamic reconfiguration, federation, legacy problems and distributed systems management.

The ODP reference model has been important in providing a usable distributed systems architecture. Significant features of ODP include object-based specification and programming, and the use of transparencies to hide aspects of distribution and viewpoints. The latter provides a basic separation of concerns, enabling different participants to observe the system from suitable perspectives and at suitable levels of abstraction. It is a central device for structuring and managing the complexity inherent in describing systems.

ODP uses five predefined viewpoints – the enterprise viewpoint, the information viewpoint, the computational viewpoint, the engineering viewpoint and the technology viewpoint. They each represent a system from one particular perspective, and these perspectives are at potentially different levels of abstraction. For example, the computational viewpoint is concerned with the algorithms and data flow of the distributed system function. It represents the system and its environment in terms of objects that interact by the transfer of information via interfaces. The engineering viewpoint, on the other hand, is concerned more with distribution mechanisms and defines the building blocks that can be combined to provide the system's functionality.

ODP is not prescriptive about the choice of specification language to be adopted with particular viewpoints. However, it does advocate the use of formal techniques to enable the precise description of distributed systems requirements. In one sense, this book is about how to do this.

Furthermore, because of the wide-ranging nature of the ODP standard, the techniques are relevant not just to ODP and its particular viewpoints, but to a whole range of specification approaches where distribution and reliability are important.

A major theme of the book is how to formally handle the new requirements aris-

ing from object-oriented distributed systems, for example, dynamic reconfiguration, encapsulation, subtyping, inheritance, real-time aspects, and so on. These may be supported either by enhancing existing notations, like LOTOS, SDL and Z, or by defining new notations, for example, Actors and Pi-calculus. This book describes both approaches in some depth.

Part One provides the context to the technical discussions in later parts. The part begins by discussing current issues in distributed systems, briefly reviewing the available techniques. The second chapter focuses on one such technique, namely, ODP, which it offers as a framework for the construction of modern distributed systems. Finally, an extrapolation is made from the issues highlighted in the first two chapters, and Chapter 3 discusses requirements on the use of formal methods.

Part Two provides a survey of formal description techniques. It considers a collection of related techniques and shows how they support object-oriented concepts. Here finite-state machine techniques such as SDL, process calculi such as E-LOTOS and state-based approaches such as Object-Z are introduced. In addition to these formal notations, semi-formal languages have an important role to play in specification and design, and to this end Part Two also provides an introduction to UML.

Part Three covers dynamic reconfiguration, looking at the different approaches of the Pi-calculus and the Actor model of computation. A discussion of the ambient calculus is also included. This provides an alternative means for describing the movement of processes and devices, including movement through administrative domains.

Part Four considers the issues of subtyping and inheritance. First, the motivation for subtyping and inheritance is considered. In particular, the use of matching in trading, binding, reuse, the role of inheritance vs subtyping and the ideas of incremental development are discussed. Then two important subtyping approaches are considered – state-based behavioural subtyping and subtyping in a process algebra setting.

Part Five provides an introduction to concurrent object-oriented languages. The first chapter in this part discusses reflection as the basis of open implementations, that is, systems that can adapt their implementation behaviour in a disciplined manner. This is followed by a discussion of the role of inheritance in a concurrent object-oriented setting.

Part Six is concerned with the support needed to express nonfunctional requirements of distributed systems using formal notations. Earlier in the book we will have considered the nature of such requirements and introduced key elements of multimedia and quality of service (QoS). This part introduces in more depth what QoS is and discusses QoS management, adaptation, specification, verification, and so on. The issues of real-time specification and validation are particularly focussed on.

Finally, in **Part Seven** the role of development architectures, composition and

viewpoints in the construction of distributed systems is considered. In particular, both composition in programming languages as well as specification architectures and viewpoints in distributed systems frameworks are discussed.

Using This Book

This book is intended to be of use to researchers in the field of distributed systems and formal methods. We hope that the book will be accessible to those new to research, for example, PhD students, while providing a standard reference book for those already established in their field. The book aims to be equally relevant to researchers in industry as well as those in the academic sector.

Several chapters in the book are based on material taught in an advanced Masters course in distributed systems, and the different chapters could be used to support MSc courses in a variety of different ways.

Acknowledgements

We would like to acknowledge the efforts of all of the contributors to this book, many of whom provided detailed and accurate texts at very short notice.

The idea for this book was conceived at the IFIP FMOODS (Formal Methods for Open Object-Based Distributed Systems) conference held at the University of Kent at Canterbury in 1997. Thanks are due in particular to Elie Najm, who, along with Jean-Bernard Stefani, initiated the series, and to all those who helped organise the 1997 FMOODS conference at the University of Kent at Canterbury.

The Networks and Distributed Systems group at Kent has over the years provided a conducive environment in which to do research and explore some of the ideas in this book, and we would like to thank in particular Peter Linington, Eerke Boiten and Maarten Steen.

We also wish to thank Lauren Cowles and her team at Cambridge University Press for their support and advice during the production of this book. Janet Bayfield at the University of Kent at Canterbury also provided valuable secretarial assistance during production of the camera-ready copy of the book.

Howard Bowman
John Derrick
Canterbury, England