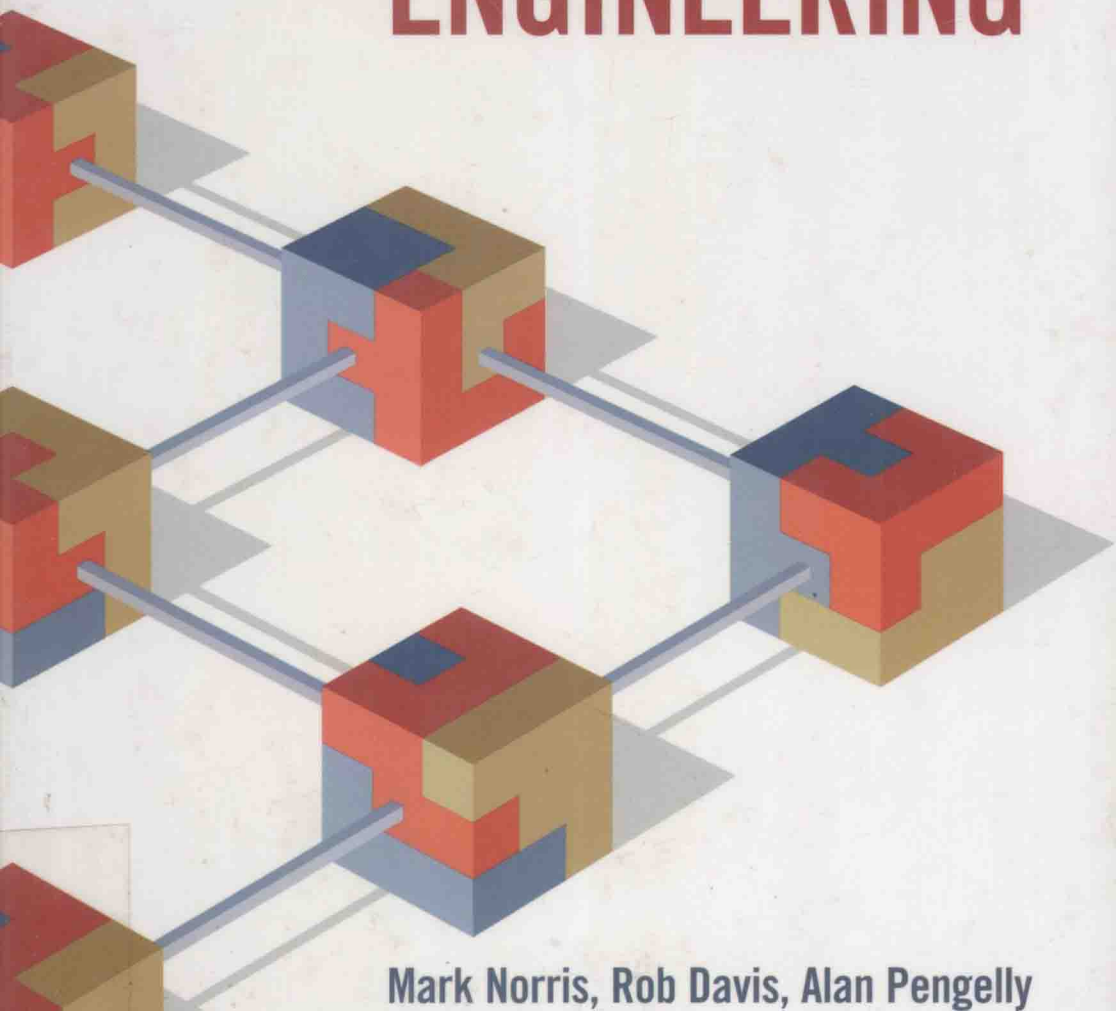


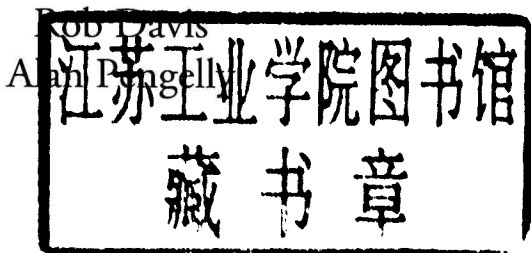
COMPONENT-BASED NETWORK SYSTEM ENGINEERING



Mark Norris, Rob Davis, Alan Pengelly

Component-Based Network System Engineering

Mark Norris



Artech House
Boston • London

Library of Congress Cataloging-in-Publication Data

Norris, Mark.

Component-based network system engineering / Mark Norris, Rob Davis, Alan Pengelly.

p. cm. — (Artech House telecommunications library)

Includes bibliographical references and index.

ISBN 1-58053-008-7 (alk. paper)

1. Component software. 2. Systems engineering. 3. Computer networks.

I. Davis, Robert E., 1950- II. Pengelly, Alan. III. Title. IV. Series

QA76.76.C66 N67 2000

004.6—dc21

99-052405

CIP

British Library Cataloguing in Publication Data

Norris, Mark

Component-based network system engineering. — (Artech House telecommunications library)

1. Computer network architectures 2. Systems engineering

I. Title II. Davis, Rob III. Pengelly, Alan

004.6'5

ISBN 1-58053-008-7

Cover design by Igor Valdman

© 2000 ARTECH HOUSE, INC.

685 Canton Street

Norwood, MA 02062

All rights reserved. Printed and bound in the United States of America. No part of this book may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without permission in writing from the publisher.

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Artech House cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

International Standard Book Number: 1-58053-008-7

Library of Congress Catalog Card Number: 99-052405

10 9 8 7 6 5 4 3 2 1

Component-Based Network System Engineering

For a listing of recent titles in the *Artech House Telecommunications Library*,
turn to the back of this book.

*To Kate, Amy, and Adam for not deleting my text files,
when it would have been just so easy. (MTN)*

Preface

The pace of change in networks and systems seems to get ever faster. The speed with which new and innovative solutions can be assembled is increasingly what differentiates the best from the rest. Hence the interest in component-based engineering. The ability to combine hardware, software, and network elements in a predictable way is no easy task, but the rewards are considerable.

It is important not only to know what the techniques are, but also what you can do with them and what is on offer to meet your needs. Where do pragmatism, standards, and theory help and how can you make them work in your environment?

This book has several unique features:

- It puts a wide range of issues into useful context. Rather than explaining the details of one area, it explains interfaces, integration, components, and architecture and shows how they relate to one another and what you need to know to capitalize on them. The aim is to provide a how-to guide to building systems that is based on firm theory and broad experience.
- It takes the pragmatic view of a complex area that has come to be dominated by technology, not always in the user's best interests. Care has been taken here to abstract from this complexity and make the topics covered accessible and relevant to real needs. The focus is on practical application, rather than technology per se. The underlying theory is not ignored—it is simply taken as a support to the endgame, rather than an end in itself.

This book is:

- Essential reading for those engaged in the construction, design, and implementation of networked systems; this book provides the broad understanding required to avoid expensive mistakes.
- A valuable professional updating guide for network designers, systems integrators, technical architects, telecommunications engineers, system analysts, and software designers, as well as business and information planners.
- A useful text for final year and postgraduate students in computer science, electrical engineering, and telecommunications courses.

It seems likely that there will be few speed limits on the information superhighways—and no turning back. Those who choose to stay in the slow lane will be left behind very quickly. Those who choose to compete in the new age need to be aware of what lies ahead. Informed choices, made now, will pay handsome dividends as complexity and choice (inevitably) rise. The end of the twentieth century is likely to be seen, in retrospect, as the adapt or atrophy period for many organizations—this book can inform an exciting but perilous journey.

A User’s Guide to This Book

This book was really inspired by our wish to give a straightforward account of a fragmented topic. We spent a long time ourselves making sense of systems, components, interfaces, and integration and would like to spare others from this subtle form of torture. So we have tried our best to cater to a wide range of tastes by explaining both the basic ideas and how they fit together.

Different parts of the book will, no doubt, be more or less relevant to different people. Some parts have been written to outline general principles, others to recount a specific technique. To help you select a suitable path through the book, here is our summary of the joys that we think each chapter contains.

	Technical Content	General Interest	Specialist Detail
Chapter 1	*	****	*
Chapter 2	**	***	**
Chapter 3	***	***	**
Chapter 4	***	***	**
Chapter 5	***	***	**
Chapter 6	***	***	***

Chapter 7	****	***	****
Chapter 8	***	***	****
Chapter 9	***	***	***
Chapter 10	*	****	*
Appendix A	*****	*	*****
Appendix B	***	**	****
Glossary	*	*	****

To help those who prefer an occasional dip into a technical book, rather than a concerted attack, we have appended a fairly large glossary that should get you through the more challenging sections.

Acknowledgments

The authors would like to thank a number of people whose help and cooperation have been invaluable, including those kind individuals who contributed ideas, advice, words, and pictures and even volunteered (we use this word in its loosest sense) to review early drafts: Professor Darrel Ince, Professor Martyn Sheppard, Dr. Alan O'Callaghan, and Ray Lewis and his colleagues in the BT Systems Integration team. Their observations, illustrative stories, guidance, and constructive criticism have always been valuable and have done much to add authority, interest, and balance to the final product.

We would also like to particularly acknowledge the contribution of the BT Component Systems Engineering Team whose ideas and experience in this exciting and challenging area underpin much of what is presented in this book. In particular we thank: Richard Shortland, Mike Scott, Andy Kelly, Alec Edwards, Greg Howett, Jim Hutton, and Philip Williams.

Finally, thanks are due to our many friends and colleagues in the telecommunication and computing industries, standards bodies, and professional organizations whose experience, advice, and inside knowledge have been invaluable.

Contents

	Preface	xvii
	A User's Guide to This Book	xviii
	Acknowledgments	xix
1	At the Edge of Communication	1
	A Global Issue	2
	Some Basic Definitions	3
	<i>An Interface</i>	3
	Some Types of Interface	4
	Why Interfaces Matter	5
	Interfaces Enable Integration	6
	Plug and Play?	7
	The Real World	8
	Tools for Integration	9
	About This Book	10
	Summary	11
	Selected Bibliography	11
2	Interfaces and Complexity	13
	Types of Interface	14
	<i>Protocols</i>	15

<i>Application Program Interface</i>	16
<i>Middleware</i>	18
<i>Remote Procedure Call</i>	19
<i>Messaging</i>	20
The Right Interface	20
The Design Problem	21
The Interface Equation	22
Summary	24
Selected Bibliography	24
3 Components	25
Components—So What Is New?	26
Why Use Components?	28
What Is a Component?	29
<i>An Example: Components in Car Assembly</i>	30
What Do We Design First—System or Components?	32
<i>Car Assembly Revisited</i>	33
<i>Personal Computer Assembly</i>	36
Bits and Pieces: What Have We Learned So Far	37
Views of Components	39
<i>Software Components: At Last!</i>	39
Generic Types of Software Components	44
<i>Black Box Components</i>	44
<i>White Box Components</i>	45
<i>Glass Box Components</i>	45
<i>Gray Box Components</i>	46
Examples of Specific Types of Software Components	47
<i>Procedures, Subroutines, Objects, and Class Libraries</i>	47
<i>Objects and Class Libraries</i>	47
<i>Operating Systems</i>	48
<i>Databases and Spreadsheets</i>	48

<i>Plugable Components</i>	50
Logical and Physical Components	52
Why Aren't Software Components the Same as Hardware Components?	52
A Layered Model of Components	54
<i>Component Technologies</i>	55
<i>Component Object Model (COM)</i>	56
<i>Common Object Request Broker Architecture (CORBA)</i>	56
<i>Enterprise Java Beans</i>	57
<i>Software Components</i>	58
<i>Mid-Tier Components</i>	58
<i>System-Level Components</i>	60
<i>Business-Level Components</i>	63
<i>Enterprise Components</i>	63
<i>Commercial Off the Shelf (COTS)</i>	64
Summary	65
Reference	67
4 Integration	69
What Is Integration?	69
<i>Dealing With Existing Installations</i>	70
<i>Stovepipe Designs</i>	70
<i>Multiple Access Requirements</i>	71
<i>Closed Interfaces</i>	73
<i>Diverse Data</i>	73
<i>Nonstandard Users</i>	74
<i>Loose and Tight Integration</i>	74
Key Concepts	75
<i>Scrap</i>	76
<i>Trap</i>	76
<i>Map</i>	76
<i>Wrap</i>	76
<i>Unwrap</i>	77

The System Integration Process	78
<i>Requirements</i>	78
<i>Design</i>	79
<i>Development</i>	79
<i>Acceptance</i>	83
<i>Build</i>	83
<i>Test</i>	83
<i>Release</i>	83
<i>Installation</i>	88
<i>Operation</i>	88
<i>Cease</i>	88
Practical Integration	88
<i>Publish a Strategy</i>	92
<i>Define an Architecture</i>	92
<i>Define an Evolution Plan</i>	93
<i>Prescribe Technology</i>	93
Summary	94
References	95
5 Architecture and Structure	97
Would You Buy a Car From This Man?	97
Sum of the Parts	98
Structured Flexibility	101
Architecture or Design?	105
Architectural Styles	108
Views of Views	109
Architectural Layers	110
Other Architectural Models	112
<i>Layers of Generality</i>	112
<i>Butler Model</i>	113
<i>The Rational 4+ 1 View</i>	114
<i>Architectural Structures</i>	117
<i>The Open Group Architectural Framework (TOGAF)</i>	118

Domain-Specific Architecture	119
Architectural Description Languages	119
Structure in Architectures	120
<i>Patterns</i>	120
<i>Anti-patterns</i>	123
<i>Pattern Systems and Languages</i>	123
<i>Frameworks</i>	125
<i>Product Lines and Product Families</i>	128
<i>Stovepipe Architectures</i>	130
Business Component Architectures	131
<i>Pipes, APIs, XML, and XMI</i>	133
<i>ERP Systems</i>	134
<i>The Open Applications Group (OAG)</i>	135
<i>Business Frameworks</i>	136
Business Objects	138
Choosing an Architecture	139
Evaluating an Architecture	140
Using an Architecture	141
The Role of Architects	143
The Shape of Things to Come	146
Summary	147
References	148
6 By Threes and Fours—Bringing It All Together	151
Motivation	153
The Scenario	164
<i>Interfaces</i>	165
<i>Networks</i>	165
<i>Server Technology</i>	166
<i>Desktop Technology</i>	167
<i>Processes</i>	167
Information Systems Integration	168
Project Management	169

<i>Phase I—Project Preparation</i>	169
<i>Phase II—Solution Design</i>	169
<i>Phase III—Detailed Design</i>	170
<i>Phase IV—Integration and Testing</i>	170
<i>Phase V—Installation and Training</i>	170
Summary	171
References	176
7 Engineering the Vision	179
The Network	180
<i>LAN</i>	181
<i>WAN</i>	184
Connection Control	184
<i>Switching</i>	186
<i>Message Switching</i>	186
<i>Packet Switching</i>	187
Network Architectures	187
<i>Peer-to-Peer Networks</i>	187
<i>Server-Based Networks</i>	188
<i>Network Operating Systems</i>	189
Network Applications	190
Interface Software and Standards	190
<i>Application Programming Interfaces</i>	193
<i>CORBA</i>	195
<i>Three-Tier Architectures</i>	198
<i>DCE</i>	198
<i>COM/DCOM</i>	199
What Should I Use?	200
Summary	201
References	202
8 Towards Component-Based Engineering	203
Component-Based Development	204
<i>System Specification, Analysis, and Design</i>	205

<i>Production Engineering</i>	222
<i>Component Design, Building, and Procurement</i>	229
<i>Systems Assembly and Delivery</i>	237
<i>Component Maintenance</i>	242
Rapid Application Development and CBD	247
Methods for CBD	248
<i>Structured Methods</i>	250
<i>OO Methods</i>	252
<i>CBD Methods</i>	253
A Generic Approach to CBD	257
<i>The Need for an Overall Approach</i>	257
<i>An Experiment in Component Approaches</i>	258
<i>Case Study Results</i>	259
<i>A Process Framework for CBD</i>	260
<i>Butler Forum Universal Component Concepts Initiative</i>	261
Describing Components	262
Repositories, Indexing, and Retrieval	264
Funding Component Approaches	265
Successful Use of Components	266
<i>Business Issues</i>	266
<i>Cultural Issues</i>	267
<i>Technical Issues</i>	269
Summary	270
Final Checklist	271
References	272

9 Interfaces and Integration—Standards and Formality	273
The Nature of Languages	276
<i>VDM Example—Creating and Removing Bank Accounts</i>	276
<i>CSP Example—A Simple Specification of the Process Diagram in Figure 9.2</i>	277

	Inherent Complexity	278
	Formality and Protocols	279
	Formal Methods	284
	Summary	287
	References	288
10	From Here to Eternity	289
	Lost Horizon	291
	Treasure Island	294
	Summary	294
	Selected Bibliography	296
Appendix A	The Interface Equation—A Formal Theory of Interfaces	297
	Introduction	298
	Formal Methods	298
	Solving the Interface Equation	300
	<i>The Discarding and Constructive Algorithms</i>	301
	<i>A Small Example</i>	307
	Conclusion	313
	References	315
Appendix B	Standards, Organizations, and Initiatives	317
	Architectural Tradeoff Analysis (ATA) Initiative	317
	Business Application Programming Interfaces (BAPI)	318
	Business Object Component Architecture (BOCA)	318
	Business Object Domain Taskforce (BODTF)	319
	Component Definition Interchange Format (CDIF)	319
	Component Definition Language (CDL)	320
	COM/DCOM	320
	CORBA	321