

---

---

# Patterns of Enterprise Application Architecture

**Martin Fowler**

With contributions from David Rice,  
Matthew Foemmel, Edward Hieatt,  
Robert Mee, and Randy Stafford

◆ Addison-Wesley

Boston • San Francisco • New York • Toronto • Montreal  
London • Munich • Paris • Madrid  
Capetown • Sydney • Tokyo • Singapore • Mexico City

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Addison-Wesley was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The author and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

The publisher offers discounts on this book when ordered in quantity for bulk purchases and special sales. For more information, please contact:

U.S. Corporate and Government Sales  
(800) 382-3419  
corpsales@pearsontechgroup.com

For sales outside of the U.S., please contact:

International Sales  
(317) 581-3793  
international@pearsontechgroup.com

Visit Addison-Wesley on the Web: [www.awprofessional.com](http://www.awprofessional.com)

*Library of Congress Cataloging-in-Publication Data*

Fowler, Martin, 1963-

Patterns of enterprise application architecture / Martin Fowler.  
p. cm.

Includes bibliographical references and index.

ISBN 0-321-12742-0 (alk. paper)

1. System design. 2. Computer architecture. 3. Application software—  
Development. 4. Business—Data processing. I. Title.

QA76.9.S88 F69 2003  
005.1—dc21

2002027743

Copyright © 2003 by Pearson Education, Inc.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher. Printed in the United States of America. Published simultaneously in Canada.

For information on obtaining permission for use of material from this work, please submit a written request to:

Pearson Education, Inc.  
Rights and Contracts Department  
501 Boylston Street, Suite 900  
Boston, MA 02116  
Fax: (617) 671-3447

ISBN 0-321-12742-0

Text printed in the United States on recycled paper at RR Donnelley in Crawfordsville, Indiana.

Seventeenth printing, July 2011

# Contents

<b>Preface</b> .....	<b>xvii</b>
Who This Book Is For .....	xx
Acknowledgments .....	xxi
Colophon .....	xxiii
<b>Introduction</b> .....	<b>1</b>
Architecture .....	1
Enterprise Applications .....	2
Kinds of Enterprise Application .....	5
Thinking About Performance .....	6
Patterns .....	9
The Structure of the Patterns .....	11
Limitations of These Patterns .....	13
<b>PART 1: The Narratives</b> .....	<b>15</b>
<b>Chapter 1: Layering</b> .....	<b>17</b>
The Evolution of Layers in Enterprise Applications .....	18
The Three Principal Layers .....	19
Choosing Where to Run Your Layers .....	22
<b>Chapter 2: Organizing Domain Logic</b> .....	<b>25</b>
Making a Choice .....	29
Service Layer .....	30
<b>Chapter 3: Mapping to Relational Databases</b> .....	<b>33</b>
Architectural Patterns .....	33
The Behavioral Problem .....	38



Reading in Data .....	40
Structural Mapping Patterns .....	41
Mapping Relationships .....	41
Inheritance .....	45
Building the Mapping .....	47
Double Mapping .....	48
Using Metadata .....	49
Database Connections .....	50
Some Miscellaneous Points .....	52
Further Reading .....	53
<b>Chapter 4: Web Presentation .....</b>	<b>55</b>
View Patterns .....	58
Input Controller Patterns .....	61
Further Reading .....	61
<b>Chapter 5: Concurrency (<i>by Martin Fowler and David Rice</i>).....</b>	<b>63</b>
Concurrency Problems .....	64
Execution Contexts .....	65
Isolation and Immutability .....	66
Optimistic and Pessimistic Concurrency Control .....	67
Preventing Inconsistent Reads .....	68
Deadlocks .....	70
Transactions .....	71
ACID .....	71
Transactional Resources .....	72
Reducing Transaction Isolation for Liveness .....	73
Business and System Transactions .....	74
Patterns for Offline Concurrency Control .....	76
Application Server Concurrency .....	78
Further Reading .....	80
<b>Chapter 6: Session State.....</b>	<b>81</b>
The Value of Statelessness .....	81
Session State .....	83
Ways to Store Session State .....	84
<b>Chapter 7: Distribution Strategies .....</b>	<b>87</b>
The Allure of Distributed Objects .....	87
Remote and Local Interfaces .....	88

Where You Have to Distribute .....	90
Working with the Distribution Boundary .....	91
Interfaces for Distribution .....	92
<b>Chapter 8: Putting It All Together .....</b>	<b>95</b>
Starting with the Domain Layer .....	96
Down to the Data Source Layer .....	97
Data Source for <i>Transaction Script</i> (110) .....	97
Data Source <i>Table Module</i> (125) .....	98
Data Source for <i>Domain Model</i> (116) .....	98
The Presentation Layer .....	99
Some Technology-Specific Advice .....	100
Java and J2EE .....	100
.NET .....	101
Stored Procedures .....	102
Web Services .....	103
Other Layering Schemes .....	103
<b>PART 2: The Patterns .....</b>	<b>107</b>
<b>Chapter 9: Domain Logic Patterns .....</b>	<b>109</b>
Transaction Script .....	110
How It Works .....	110
When to Use It .....	111
The Revenue Recognition Problem .....	112
Example: Revenue Recognition (Java) .....	113
Domain Model .....	116
How It Works .....	116
When to Use It .....	119
Further Reading .....	119
Example: Revenue Recognition (Java) .....	120
Table Module .....	125
How It Works .....	126
When to Use It .....	128
Example: Revenue Recognition with a Table Module (C#) ....	129
Service Layer ( <i>by Randy Stafford</i> ) .....	133
How It Works .....	134
When to Use It .....	137

Further Reading .....	137
Example: Revenue Recognition (Java) .....	138
<b>Chapter 10: Data Source Architectural Patterns .....</b>	<b>143</b>
Table Data Gateway .....	144
How It Works .....	144
When to Use It .....	145
Further Reading .....	146
Example: Person Gateway (C#) .....	146
Example: Using ADO.NET Data Sets (C#) .....	148
Row Data Gateway .....	152
How It Works .....	152
When to Use It .....	153
Example: A Person Record (Java) .....	155
Example: A Data Holder for a Domain Object (Java) .....	158
Active Record .....	160
How It Works .....	160
When to Use It .....	161
Example: A Simple Person (Java) .....	162
Data Mapper .....	165
How It Works .....	165
When to Use It .....	170
Example: A Simple Database Mapper (Java) .....	171
Example: Separating the Finders (Java) .....	176
Example: Creating an Empty Object (Java) .....	179
<b>Chapter 11: Object-Relational Behavioral Patterns .....</b>	<b>183</b>
Unit of Work .....	184
How It Works .....	184
When to Use It .....	189
Example: <i>Unit of Work</i> with Object Registration (Java) <i>(by David Rice)</i> .....	190
Identity Map .....	195
How It Works .....	195
When to Use It .....	198
Example: Methods for an <i>Identity Map</i> (Java) .....	198

Lazy Load .....	200
How It Works .....	200
When to Use It .....	203
Example: Lazy Initialization (Java) .....	203
Example: Virtual Proxy (Java) .....	203
Example: Using a Value Holder (Java) .....	205
Example: Using Ghosts (C#) .....	206
<b>Chapter 12: Object-Relational Structural Patterns .....</b>	<b>215</b>
Identity Field .....	216
How It Works .....	216
When to Use It .....	220
Further Reading .....	221
Example: Integral Key (C#) .....	221
Example: Using a Key Table (Java) .....	222
Example: Using a Compound Key (Java) .....	224
Foreign Key Mapping .....	236
How It Works .....	236
When to Use It .....	239
Example: Single-Valued Reference (Java) .....	240
Example: Multitable Find (Java) .....	243
Example: Collection of References (C#) .....	244
Association Table Mapping .....	248
How It Works .....	248
When to Use It .....	249
Example: Employees and Skills (C#) .....	250
Example: Using Direct SQL (Java) .....	253
Example: Using a Single Query for Multiple Employees (Java) <i>(by Matt Foemmel and Martin Fowler)</i> .....	256
Dependent Mapping .....	262
How It Works .....	262
When to Use It .....	263
Example: Albums and Tracks (Java) .....	264
Embedded Value .....	268
How It Works .....	268
When to Use It .....	268

Further Reading .....	270
Example: Simple Value Object (Java) .....	270
Serialized LOB .....	272
How It Works .....	272
When to Use It .....	274
Example: Serializing a Department Hierarchy in XML (Java) .....	274
Single Table Inheritance .....	278
How It Works .....	278
When to Use It .....	279
Example: A Single Table for Players (C#) .....	280
Loading an Object from the Database .....	281
Class Table Inheritance .....	285
How It Works .....	285
When to Use It .....	286
Further Reading .....	287
Example: Players and Their Kin (C#) .....	287
Concrete Table Inheritance .....	293
How It Works .....	293
When to Use It .....	295
Example: Concrete Players (C#) .....	296
Inheritance Mappers .....	302
How It Works .....	303
When to Use It .....	304
<b>Chapter 13: Object-Relational Metadata Mapping Patterns .....</b>	<b>305</b>
Metadata Mapping .....	306
How It Works .....	306
When to Use It .....	308
Example: Using Metadata and Reflection (Java) .....	309
Query Object .....	316
How It Works .....	316
When to Use It .....	317
Further Reading .....	318
Example: A Simple <i>Query Object</i> (Java) .....	318



Repository ( <i>by Edward Hieatt and Rob Mee</i> ) . . . . .	322
How It Works . . . . .	323
When to Use It . . . . .	324
Further Reading . . . . .	325
Example: Finding a Person's Dependents (Java) . . . . .	325
Example: Swapping <i>Repository</i> Strategies (Java) . . . . .	326
<b>Chapter 14: Web Presentation Patterns.</b> . . . .	<b>329</b>
Model View Controller . . . . .	330
How It Works . . . . .	330
When to Use It . . . . .	332
Page Controller . . . . .	333
How It Works . . . . .	333
When to Use It . . . . .	334
Example: Simple Display with a Servlet Controller and a JSP View (Java) . . . . .	335
Example: Using a JSP as a Handler (Java) . . . . .	337
Example: Page Handler with a Code Behind (C#) . . . . .	340
Front Controller . . . . .	344
How It Works . . . . .	344
When to Use It . . . . .	346
Further Reading . . . . .	347
Example: Simple Display (Java) . . . . .	347
Template View . . . . .	350
How It Works . . . . .	351
When to Use It . . . . .	354
Example: Using a JSP as a View with a Separate Controller (Java) . . . . .	355
Example: ASP.NET Server Page (C#) . . . . .	357
Transform View . . . . .	361
How It Works . . . . .	361
When to Use It . . . . .	362
Example: Simple Transform (Java) . . . . .	363
Two Step View . . . . .	365
How It Works . . . . .	365
When to Use It . . . . .	367

Example: Two Stage XSLT (XSLT) .....	371
Example: JSP and Custom Tags (Java) .....	374
Application Controller .....	379
How It Works .....	380
When to Use It .....	381
Further Reading .....	382
Example: State Model <i>Application Controller</i> (Java) .....	382
<b>Chapter 15: Distribution Patterns .....</b>	<b>387</b>
Remote Facade .....	388
How It Works .....	389
When to Use It .....	392
Example: Using a Java Session Bean as a <i>Remote Facade</i> (Java) .....	392
Example: Web Service (C#) .....	395
Data Transfer Object .....	401
How It Works .....	401
When to Use It .....	406
Further Reading .....	407
Example: Transferring Information About Albums (Java) .....	407
Example: Serializing Using XML (Java) .....	411
<b>Chapter 16: Offline Concurrency Patterns .....</b>	<b>415</b>
Optimistic Offline Lock ( <i>by David Rice</i> ) .....	416
How It Works .....	417
When to Use It .....	420
Example: Domain Layer with <i>Data Mappers (165)</i> (Java) .....	421
Pessimistic Offline Lock ( <i>by David Rice</i> ) .....	426
How It Works .....	427
When to Use It .....	431
Example: Simple Lock Manager (Java) .....	431
Coarse-Grained Lock ( <i>by David Rice and Matt Foemmel</i> ) .....	438
How It Works .....	438
When to Use It .....	441
Example: Shared <i>Optimistic Offline Lock (416)</i> (Java) .....	441
Example: Shared <i>Pessimistic Offline Lock (426)</i> (Java) .....	446
Example: Root <i>Optimistic Offline Lock (416)</i> (Java) .....	447

Implicit Lock ( <i>by David Rice</i> ) . . . . .	449
How It Works . . . . .	450
When to Use It . . . . .	451
Example: Implicit <i>Pessimistic Offline Lock</i> (426) (Java) . . . . .	451
<b>Chapter 17: Session State Patterns . . . . .</b>	<b>455</b>
Client Session State . . . . .	456
How It Works . . . . .	456
When to Use It . . . . .	457
Server Session State . . . . .	458
How It Works . . . . .	458
When to Use It . . . . .	460
Database Session State . . . . .	462
How It Works . . . . .	462
When to Use It . . . . .	464
<b>Chapter 18: Base Patterns. . . . .</b>	<b>465</b>
Gateway . . . . .	466
How It Works . . . . .	466
When to Use It . . . . .	467
Example: A Gateway to a Proprietary Messaging Service (Java) . . . . .	468
Mapper . . . . .	473
How It Works . . . . .	473
When to Use It . . . . .	474
Layer Supertype . . . . .	475
How It Works . . . . .	475
When to Use It . . . . .	475
Example: Domain Object (Java) . . . . .	475
Separated Interface . . . . .	476
How It Works . . . . .	477
When to Use It . . . . .	478
Registry . . . . .	480
How It Works . . . . .	480
When to Use It . . . . .	482



Example: A Singleton Registry (Java) .....	483
Example: Thread-Safe <i>Registry</i> (Java) (by Matt Foemmel and Martin Fowler) .....	484
Value Object .....	486
How It Works .....	486
When to Use It .....	487
Money .....	488
How It Works .....	488
When to Use It .....	490
Example: A Money Class (Java) (by Matt Foemmel and Martin Fowler) .....	491
Special Case .....	496
How It Works .....	497
When to Use It .....	497
Further Reading .....	497
Example: A Simple Null Object (C#) .....	498
Plugin (by David Rice and Matt Foemmel) .....	499
How It Works .....	499
When to Use It .....	500
Example: An Id Generator (Java) .....	501
Service Stub (by David Rice) .....	504
How It Works .....	504
When to Use It .....	505
Example: Sales Tax Service (Java) .....	505
Record Set .....	508
How It Works .....	508
When to Use It .....	510
References .....	511
Index .....	517