

Extracted from:

Domain-Driven Design

Using Naked Objects

This PDF file contains pages extracted from Domain-Driven Design, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragprog.com>.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printer versions; the content is otherwise identical.

Copyright © 2009 The Pragmatic Programmers, LLC.

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.

The
Pragmatic
Programmers

Domain-Driven Design Using Naked Objects



Dan Haywood

Edited by Susannah Davidson Pfäzler



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 The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Our Pragmatic courses, workshops, and other products can help you and your team create better software and have more fun. For more information, as well as the latest Pragmatic titles, please visit us at

<http://www.pragprog.com>

Copyright © 2009 Dan Haywood.

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.

ISBN-10: 1-934356-44-1

ISBN-13: 978-1-934356-44-9

Printed on acid-free paper.

P1.0 printing, December 2009

Version: 2010-1-16

Contents

Preface	16
Who This Book Is For	17
How the Book Is Organized	18
Case Study and Exercises	19
Conventions	20
Further Resources	20
I Tools	22
1 Getting Started	23
1.1 Understanding Domain-Driven Design	23
1.2 The Essentials of DDD	24
1.3 Introducing Naked Objects	28
1.4 Naked Objects in About Five Minutes	30
1.5 How Naked Objects Helps with DDD	36
1.6 The Big Picture	39
2 Identifying the Domain Concepts	43
2.1 Introducing CarServ	43
2.2 Getting Ready	45
2.3 Creating the Domain Classes	47
2.4 Using Repositories to Locate Objects	51
2.5 Identifying Objects to the User	55
2.6 Capturing Simple Business Rules	61
2.7 Providing Choices for Properties	63
3 Relating Objects Together	66
3.1 Associating Objects	66
3.2 Adding Describing Concepts	73
3.3 Capturing Business Rules for Collections	79

4	Rapid Prototyping	82
4.1	Fixtures for Setting Up Domain Objects	83
4.2	Fixtures for Setting Up the Clock	86
4.3	Fixtures for Setting Up User Sessions	89
4.4	Organizing Fixtures into Hierarchies	93
5	Creating Behaviorally Complete Objects	97
5.1	Adding Behavior to Domain Objects	98
5.2	Validating Action Arguments	100
5.3	Making Actions Friendlier to Use	103
5.4	Adding Finders to Repositories	105
6	Implementing Business Rules	108
6.1	Validation Recap	109
6.2	Disabling Class Members	112
6.3	Hiding Class Members	115
6.4	Declarative Rules and the Object Life Cycle	119
6.5	Validating the Entire Object	122
7	Using Value Types	126
7.1	Identifying Value Types	127
7.2	Pushing Business Rules onto a Value Type	128
7.3	Adding a Third-Party Value Type	131
7.4	Specifying Defaults and Other Characteristics	139
8	Isolating Infrastructure Services	142
8.1	A Taxonomy of Services	143
8.2	The Domain Object Container	145
8.3	Dependency Injection	147
8.4	Using Services in Fixtures	148
8.5	Requirements for Writing Services	149
8.6	Using Interfaces for Repositories	151
8.7	Implementing a Calendar Service	153
8.8	Hints and Tips for Writing Services	156
II	Techniques	159
9	Distributing Class Responsibilities	160
9.1	Applying Coad Colors	161
9.2	Factoring Out Objects	163
9.3	Balancing Responsibilities	168
9.4	Representing Large Collections with Finder	170
9.5	Contributing Actions from Services	174

10 Applying Domain Patterns	178
10.1 Type as Factory Pattern	179
10.2 Knowledge Level Pattern	186
10.3 Null Object Pattern	189
10.4 Role Object Pattern	191
10.5 User Peer Object Pattern	196
10.6 Strategy Pattern	197
10.7 Process Object Pattern	202
11 Keeping the Model Maintainable	208
11.1 Analyzing the Structure of CarServ	209
11.2 Decoupling by Moving Responsibilities	212
11.3 Decoupling by Introducing Interfaces	214
11.4 Layering Modules	221
11.5 Decoupling by Splitting Classes	223
11.6 Introducing an Application Package	224
11.7 An Application Architecture Blueprint	227
12 Scenario Testing	231
12.1 Writing Developer Tests	232
12.2 Scenario Testing Using FitNesse	237
12.3 Getting Ready to Write Scenario Tests	238
12.4 Writing Scenario Tests	243
12.5 Hints and Tips	251
III Practices	254
13 Developing Domain Applications	255
13.1 The Layered Architecture	256
13.2 Deployment Options	257
13.3 Which Option to Choose?	260
13.4 Development Activities	263
13.5 Configuration Management	266
13.6 Working Effectively	269
14 Naked Objects as a Design Tool	273
14.1 Using Naked Objects Only in Development	273
14.2 Decoupling from the Framework	274
14.3 Programming Model Interaction Protocol	278
14.4 Changing the Programming Model	280

15 Integrating with Web Frameworks	283
15.1 Deploying an Embedded Metamodel	284
15.2 Integrating Layers with the Custom Presentation Option	295
16 Integrating with the Database	301
16.1 Configuring XML Persistence	302
16.2 Mapping Entities Using JPA Annotations	304
16.3 Mapping Value Objects Using JPA Annotations	310
16.4 Mapping Relationships	312
16.5 Porting over Repositories	316
16.6 Deploying and Running the Application	319
17 Integrating Within the Enterprise	325
17.1 Bounded Context Patterns	326
17.2 Exposing a RESTful Web Service for Other Systems	328
17.3 Integrating Using an Enterprise Service Bus	334
18 Deploying the Full Runtime	347
18.1 Deploying the Application	347
18.2 Securing the Application	356
18.3 Deploying the Sister Projects	361
18.4 A CarServ Retrospective	366
18.5 The DSFA Application	367
18.6 Closing Thoughts	368
IV Appendixes	376
A Programming Model Cheat Sheet	377
B Eclipse Templates	381
C Bibliography	385
Index	388

The Pragmatic Bookshelf

The Pragmatic Bookshelf features books written by developers for developers. The titles continue the well-known Pragmatic Programmer style and continue to garner awards and rave reviews. As development gets more and more difficult, the Pragmatic Programmers will be there with more titles and products to help you stay on top of your game.

Visit Us Online

Domain-Driven Design using Naked Objects Home Page

<http://pragprog.com/titles/dhnako>

Source code from this book, errata, and other resources. Come give us feedback, too!

Register for Updates

<http://pragprog.com/updates>

Be notified when updates and new books become available.

Join the Community

<http://pragprog.com/community>

Read our weblogs, join our online discussions, participate in our mailing list, interact with our wiki, and benefit from the experience of other Pragmatic Programmers.

New and Noteworthy

<http://pragprog.com/news>

Check out the latest pragmatic developments, new titles and other offerings.

Buy the Book

If you liked this eBook, perhaps you'd like to have a paper copy of the book. It's available for purchase at our store: pragprog.com/titles/dhnako.

Contact Us

Online Orders:	www.pragprog.com/catalog
Customer Service:	support@pragprog.com
Non-English Versions:	translations@pragprog.com
Pragmatic Teaching:	academic@pragprog.com
Author Proposals:	proposals@pragprog.com
Contact us:	1-800-699-PROG (+1 919 847 3884)