Building J2EE™ Applications with the Rational Unified Process
Building J2EE™ Applications with the Rational Unified Process

Peter Eeles
Kelli Houston
Wojtek Kozaczynski

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, Inc. was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The authors 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 special sales. For more information, please contact:

Pearson Education Corporate Sales Division
201 W. 103rd Street
Indianapolis, IN 46290
(800) 428-5331
corpsales@pearsoned.com

Visit AW on the Web: www.awprofessional.com

Library of Congress Cataloging-in-Publication Data

Copyright © 2003 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
75 Arlington Street, Suite 300
Boston, MA 02116
Fax: (617) 848-7047

ISBN 0-201-79166-8
Text printed on recycled paper
1 2 3 4 5 6 7 8 9 10—MA—0605040302
First printing, August 2002
Contents

Foreword by Philippe Kruchten xi
Foreword by John Crupi xiii
Preface xvii

Chapter 1 Introduction 1
How This Book Is Organized 2

Chapter 2 An Introduction to the Java 2 Platform, Enterprise Edition 5
Enterprise Concerns 5
   Business Concerns 6
   Integration Concerns 6
   Development Concerns 7
Multitier Architectures and the J2EE Platform 7
J2EE Platform Overview 9
   J2EE Technology Overview 10
      Containers 12
      Presentation Tier 14
      Business Tier 15
      Integration Tier 15
      J2EE Deployment Configurations 15
J2EE Component Technologies 19
   Applets 19
   Application Clients 20
Chapter 3  An Introduction to the Rational Unified Process  35
Best Practices—The Foundation of RUP  35
RUP Key Concepts  37
   Artifacts  38
   Roles  39
   Activities  40
   Disciplines  41
   Workflow Details  43
   Iterations  44
   Phases  46
RUP as a Process Framework  49
Summary  49

Chapter 4  An Introduction to the J2EE Developer Roadmap  51
What Is a RUP Roadmap?  51
J2EE Developer Roadmap—Scope and Rationale  52
J2EE Developer Roadmap—J2EE-Specific Content  57
Summary  59

Chapter 5  An Introduction to the Sample Application  61
Application Overview  61
Problem Statement  62
Stakeholders and Users  63
   Stakeholders: Their Concerns and Responsibilities  63
   Users, Their Concerns and Responsibilities  64
Functional Features and Constraints  64
Other Requirements and Properties  65
   Nonfunctional Properties  65
   User Environment  66
   Deployment Environment  66
Summary  66

Chapter 6  Requirements  67
Requirements and Iterative Development  69
Requirements Overview  69
Workflow Detail: Define the System 70
   Activity: Capture a Common Vocabulary 72
   Activity: Find Actors and Use Cases 73
   Activity: Prioritize Use Cases 81
   Activity: Review the Requirements 83
Workflow Detail: Refine the System Definition 84
   Activity: Detail a Use Case 86
   Activity: Structure the Use-Case Model 91
   Activity: Review the Requirements 92
Summary 95

Chapter 7 Analysis 97
Analysis and Iterative Development 98
Analysis Overview 99
Workflow Detail: Define an Initial Architecture 100
   Activity: Architectural Analysis 102
   Activity: Review the Initial Architecture 109
Workflow Detail: Analyze Behavior 110
   Activity: Model the User Experience 112
   Activity: Review the User Experience 123
   Activity: Use-Case Analysis 124
   Activity: Review the Analysis 134
Summary 136

Chapter 8 Design 137
Design and Iterative Development 138
Design Overview 139
Workflow Detail: Refine the Architecture 139
   Activity: Identify Design Mechanisms 141
   Activity: Identify Design Elements 145
   Activity: Incorporate Existing Design Elements 166
   Activity: Describe Distribution and Concurrency 169
   Activity: Review the Architecture 173
Workflow Detail: Detail the Design 174
   Activity: Use-Case Design 176
   Activity: Subsystem Design 179
   Activity: Component Design 185
   Activity: Class Design 186
   Activity: Database Design 189
   Activity: Review the Design 191
Summary 192
## Chapter 9 Implementation

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation and Iterative Development</td>
<td>195</td>
</tr>
<tr>
<td>Implementation Overview</td>
<td>195</td>
</tr>
<tr>
<td>Workflow Detail: Structure the Implementation Model</td>
<td>196</td>
</tr>
<tr>
<td>Activity: Structure the Implementation Model</td>
<td>198</td>
</tr>
<tr>
<td>Activity: Review the Implementation</td>
<td>203</td>
</tr>
<tr>
<td>Workflow Detail: Implement Design Elements</td>
<td>204</td>
</tr>
<tr>
<td>Activity: Implement Design Elements</td>
<td>204</td>
</tr>
<tr>
<td>Activity: Perform Unit Tests</td>
<td>214</td>
</tr>
<tr>
<td>Activity: Review the Implementation</td>
<td>216</td>
</tr>
<tr>
<td>Summary</td>
<td>217</td>
</tr>
</tbody>
</table>

## Chapter 10 Additional Topics

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Modeling</td>
<td>219</td>
</tr>
<tr>
<td>Testing</td>
<td>220</td>
</tr>
<tr>
<td>Deployment</td>
<td>220</td>
</tr>
<tr>
<td>Configuration and Change Management</td>
<td>221</td>
</tr>
<tr>
<td>Project Management</td>
<td>222</td>
</tr>
<tr>
<td>Environment</td>
<td>223</td>
</tr>
<tr>
<td>Other Considerations</td>
<td>224</td>
</tr>
<tr>
<td>In Conclusion–a Note from the Authors</td>
<td>225</td>
</tr>
</tbody>
</table>

## Appendix A Describing a Software Architecture

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>227</td>
</tr>
<tr>
<td>What Is Architecture?</td>
<td>227</td>
</tr>
<tr>
<td>Communicating the Architecture of a System</td>
<td>228</td>
</tr>
<tr>
<td>The Software Architecture Document</td>
<td>230</td>
</tr>
</tbody>
</table>

## Appendix B Modeling Conventions

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>UML Representation of J2EE Developer</td>
<td>233</td>
</tr>
<tr>
<td>Roadmap Elements</td>
<td>233</td>
</tr>
<tr>
<td>Requirements Artifacts</td>
<td>234</td>
</tr>
<tr>
<td>Analysis Artifacts</td>
<td>234</td>
</tr>
<tr>
<td>Design Artifacts</td>
<td>235</td>
</tr>
<tr>
<td>Implementation Artifacts</td>
<td>236</td>
</tr>
<tr>
<td>Design Relationships</td>
<td>237</td>
</tr>
<tr>
<td>Design Class Attributes and Operations</td>
<td>237</td>
</tr>
<tr>
<td>J2EE Developer Roadmap Model Structure</td>
<td></td>
</tr>
<tr>
<td>Guidelines</td>
<td>238</td>
</tr>
<tr>
<td>Use-Case Model</td>
<td>238</td>
</tr>
<tr>
<td>User-Experience Model</td>
<td>238</td>
</tr>
<tr>
<td>Design Model</td>
<td>239</td>
</tr>
<tr>
<td>Implementation Model</td>
<td>243</td>
</tr>
</tbody>
</table>
Foreword

For a long time, I've wanted a book that clearly describes a new technology, a process, and a related project so that I can understand the practical relationship between these essential ingredients that all software development organizations juggle with. But I also understand how difficult it can be to combine all these elements in one book.

Describing a new technology is hard. Well, sort of. If you are an expert, you can indeed describe it to any level of detail, enumerate all the parts, and produce yet another one of those massive 1,500-page volumes that seem to be the craze in the IT world lately. But then it reads like a catalog of integrated circuits—very dry and boring, with few "why's" and "how's" to relate the parts. In fact, you want easy access to this information, but you do not want to read it as a book. It is hard to understand how to effectively use a new technology without putting it in the context of a project and a process, that is, without giving examples of recipes for how to actually use it.

Describing a software development process is hard. I know. This is what I have tried to do for the last six years. Either the process is too generic, painted in broad strokes, and the reader is left to fill in the methods, recipes, templates, and heuristics required to make project decisions. Or, the process is overly precise and specific, and gives in great detail the methods, recipes, templates, and heuristics for specific domains and technologies, and then the process becomes huge, unwieldy, and cumbersome—a straitjacket. A process only has value in as much as it helps the practitioners develop software. Without ties to specific examples, it remains abstract and mysterious. Without ties to technologies, too
much is left for the practitioner to invent. It is a classic double-bind: cursed if you do, cursed if you don’t.

Describing the inner workings of one specific software project is moderately interesting in itself, unless you are its developing or maintaining organization. After all, the likelihood that your project will use exactly the same solutions as someone else’s is low. Giving an example alone, without the reasoning for the choices behind it, is of little value. Also, many textbook examples are so small that their illustrative value is limited, which sometimes even turns against the features that they want to illustrate.

But if we could combine the essence of a technology, illustrate its use on an actual, full-size example project, and at the same time communicate the development process that was used to design this project, we would win on all three fronts. We put the technology to practical use: the parts of the catalog will now make sense. By going through the process, we understand how and why the decisions were made to pick this or that part, which is more important than the decision itself because you want to replicate the process, not just copy the solutions. By using a single, nontrivial example, you expose some of the harder problems, the ones related to the complexity of the system, its architecture, rather than the shortsighted details of one single feature of the technology.

Combining in one single book these three aspects:

- Technologies. The UML, Java, J2EE
- Process. The Rational Unified Process (RUP)
- Example. The Online Auction application

is precisely what my friends and colleagues Kelli, Peter, and Wojtek have successfully attempted here, without creating yet another monster of paper. I am personally delighted that they have picked the RUP as the basis for the process aspect. Like the puppets in The Nutcracker ballet, the RUP in this book suddenly comes to life. And, the book gives a new perspective to RUP and a rationale (no pun intended) for all the RUP activities, templates, and heuristics that they have selected. The book also helps the reader understand that the RUP is very accessible and flexible, and not a heavyweight bureaucratic monster.

This is a book that a J2EE developer can use as a compass for direction and as a jump-off point to go elsewhere for details. Inside I learned quite a few things about J2EE—a technology that I have never used on an actual project; I learned a lot through the example of the auction system. And at the same time, although I am deeply familiar with the RUP, I now see it in a different light. I am sure you will enjoy and learn a lot from this book, and that it will help you understand not only the technology that you will want to use on your next soft-
ware endeavor, but also the “how’s” and the “why’s” to put it to work. Kelli, Peter, Wojtek: you’ve added 1 + 1 + 1, and got a 10. Bravo! And thank you.

Philippe Kruchten
Rational Fellow
Director, RUP Development
Foreword

I did a search on Amazon.com for all books with Java in their title and came up with almost 1,700 books. That is pretty amazing for a technology that isn’t even a decade old. So I took a look at some of the titles and not surprisingly, most of the books were “how to” learn some Java technology. Very few were actually about “how to” design and architect with Java technology. This makes sense, because just as you first must learn how to walk before you can run, you (in my opinion) must first learn the technology before you can even think about designing with the technology.

When writing *Core J2EE Patterns*, Deepak Alur, Dan Malks, and I added an epilogue on pattern-driven design (“J2EE Patterns Applied”). Our original intent was to show how to go from a business problem to an application design that was based on the patterns. We began documenting a prescriptive process for selecting the appropriate J2EE patterns. But, as we got further into documenting this process, we realized that this chapter was growing to the size of all the previous chapters. Also, based on early feedback from reviewers, we were told that for the context of a patterns book, developers did not want to see a lot of process; they wanted more of a learn-by-example approach. This was fine for us, because we weren’t methodologists and were happy to leave that to the experts.

Lucky for us, this book was written. Kelli, Peter, and Wojtek have done a great job of providing process understanding in the context of J2EE. The authors provide a prescriptive model with input artifacts, activities, and resulting artifacts. Using a “learn-by-example” approach, they show in an easily understandable
and digestible way how process is applied to build J2EE applications. Once you start getting the hang of things and become comfortable with RUP, it is no longer necessary to constantly refer to it for guidance—it has become “second nature.” In this case, we can treat the process as being descriptive in that it is a rich source of knowledge should we need it.

Since the same process can be treated as being either “prescriptive” or “descriptive,” this book can be used by both novices and experts, and anyone in between.

Technology is not getting any easier; actually I think it is getting more sophisticated and complex. The end result is that new technology does more for you and essentially lets you create much more complex systems. J2EE can therefore be viewed as a set of technologies and APIs, which essentially gives you a platform on which to design and build sophisticated business applications—but, how? In the typical cycle of new technologies, you end up spending the early part of the cycle learning the technology. Then, you move on to learning how to best apply the technology and how not to apply the technology. This part of the cycle can take a few years to really figure out. Unfortunately, early adopters have little or no guidance in the form of best practices, patterns, and so on—basically because it hasn’t been practiced much.

From the success of our Core J2EE Patterns book, we know that developers like J2EE patterns. Patterns seem to be the “sweet spot” of design reuse and at the right abstraction for developers. But, there is great opportunity to take patterns beyond the cataloging in a book. There is the opportunity to automate the application of the patterns and to introduce patterns into a process. And that is exactly what the authors have done with RUP and J2EE. RUP is a very powerful process and has been well defined and streamlined for J2EE in this book. What I really like about this book is how the authors introduce J2EE patterns into the RUP design process. The authors have also recognized that patterns by themselves are of modest value. The real value appears when the patterns are combined to address a larger problem. This is what the authors refer to as “reference architectures” and, in my opinion, really help you jump-start and accelerate J2EE design and development.

I think this book is a great asset to all J2EE developers who want to take advantage of RUP and learn how to apply a proven methodology to the J2EE design and development process.

John Crupi  
Coauthor Core J2EE Patterns  
Distinguished Engineer  
Sun Microsystems
Preface

About This Book

This book is about developing Java 2 Platform, Enterprise Edition (J2EE) applications with the Rational Unified Process (RUP) and the Unified Modeling Language (UML). There are a number of books describing the J2EE platform and its associated technologies. There are also books about software development processes and about RUP, in particular. However, we could not find a book that described and exemplified how to use a subset of RUP that contains only the most relevant parts of an otherwise large and comprehensive process, to build J2EE applications. The book bridges this gap between the RUP and J2EE technologies.

A world that combines J2EE, RUP and UML is complex. This world requires a good map if you are to successfully navigate your way through it. This book provides such a map both metaphorically and literally. It is metaphorical because the very objective of the book is to provide a description of a "safe path" through the J2EE application development activities. It is also literal because we define a "J2EE Developer Roadmap," which is a process map, to guide us. We describe the content of the book in more detail in Chapter 1, Introduction.

Who This Book Is For

While writing the book we kept three audiences in mind. The first audience is software architects, designers and developers familiar with the J2EE platform...
and its technologies, wanting to understand how to apply them in the context of a software development process. The second audience is development team members familiar with RUP, looking for help in applying the process to J2EE development. The third audience is software professionals who are neither familiar with J2EE nor with RUP, and are looking for an example of how to use RUP in the development of a J2EE application.

Conventions Used in This Book

The book is roughly divided into two parts. The first part, Chapters 2–5, summarizes the J2EE technologies, introduces RUP and the J2EE Developer Roadmap, and briefly describes our sample application. The rest of the book is what we refer to as the “process chapters” and contains a guided tour through a J2EE application development project.

We structured the process chapters to make them easy to read “at a glance,” and to make it easy to refer to specific topics of interest. Each process chapter is organized primarily by activities. We provide diagrams that indicate the overall flow between the activities and the key artifacts that are consumed and produced by the activities. For each activity, we provide an overview table that summarizes the purpose of the activity, its input and output artifacts, and a summary of the steps performed within the activity. Following the table, we then describe the details of each of these steps, using the sample application.

Also in the process chapters we used a few styles and conventions. We capitalized all references to RUP process elements such as roles, activities, steps and artifacts.

Acknowledgments

We would first like to thank Rational Software Corporation, who has allowed us to incorporate various aspects of Rational’s intellectual property in this book (specifically, the elements of the Rational Unified Process). We would also like to express our thanks to the founders and employees of Rational for building a company whose mission, core values, and culture encourage the writing of books such as this.

A special thanks goes to our team of reviewers. The breadth and depth of their collective experience has helped us immensely in ensuring that this book provides the right focus and, more importantly, is of enduring value. We would like to thank Kevin Benner, Grady Booch, John Cheesman, Jim Conallen, John Crupi, Kevin Kelly, Philippe Kruchten, Boris Lublinsky, Bruce MacIsaac, Jim Ning, Davyd Norris, Oliver Sims, Dave West, and Andy Winskill.
Grady, by the way, is also responsible for sowing the seed from which this book has grown. Should Grady ever see a presentation of yours and suggest you “write a book,” then know that he’s serious. Grady—thanks for the support and encouragement from start to finish.

This book has benefited from the experiences of, and support given by, customers, Rational partners, colleagues, and peers alike. In this respect, we would like to thank Alan Brown, Christina Cooper-Bland, Ian Forsythe, Manjinder Gahir, Bob Houston, Russell Norlund, Alan Perkins, and Jim Thario.

We would also like to thank everyone at Addison-Wesley involved in the production of this book. In particular, we would like to thank our editor, Paul Becker, for his guidance and advice.

Finally, and most importantly, we would like to thank our families. Without their weekends and their evenings, this book would simply not exist.

To my wife Karen and our children, Daniel, Thomas, and Christopher, for your constant support and understanding. I would like to dedicate this book to my mother Nancy and my father Bob.—PE

To my husband Bob, a respected colleague and friend, for his unwavering support and encouragement; and to our children, Katherine and Ryan, whose enthusiasm and zest for life inspire me every day.—KH

I would like to dedicate this book to my wife Jola and my children Tommy and Natasza. Thank you for your understanding and support.—WK