Agile Business Rule Development

Jérôme Boyer • Hafedh Mili

Agile Business Rule Development

Process, Architecture, and JRules Examples

Springer

Mr. Jérôme Boyer
IBM
4400 North First Street
San Jose, CA, 95134
USA
boyerje@us.ibm.com

Prof. Hafedh Mili
Université du Québec á Montréal
Dépt. Informatique
C.P. 8888
Succursale centre-ville
Montréal Québec H3C 3P8
Canada
hafedh.mili@uqam.ca

ISBN 978-3-642-19040-7 e-ISBN 978-3-642-19041-4
DOI 10.1007/978-3-642-19041-4
Springer Heidelberg Dordrecht London New York

ACM Computing Classification (1998): J.1., H.3.5, I.2, D.2

Library of Congress Control Number: 2011924779

Cover design: KuenkelLopka GmbH

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To my family and friends who support my day to day work and humor

To Amel, Haroun and Khalil, for giving a meaning to what I do

To Aicha, Taieb, Faouzi, Ali, Kamel, Fatma, Hedia, Seloua, Sadok and Nebiha, for being there when it matters

To Lal-Melika and in memory of Si El Moncef

Foreword I

We all make a huge variety of decisions every day. For the vast majority of our daily chores we make those decisions based on the set of experiences and philosophies that we have developed and evolved over time. It is that combination of experience that makes us who we are—and that ensures that we are uniquely qualified to perform our jobs. The problem is that all too often the things that make us unique will lead us to making different decisions from everyone else in the organization. Those differences may be acceptable for a large class of the decisions we make. However, that can also be detrimental to the organization when it comes to certain *core* business processes.

Businesses that are able to capture the criteria by which they make business decisions are able to drive better business results. By capturing those criteria you can reason about their effectiveness. You can combine the best of everyone's experience to ensure you are able to respond to the most comprehensive set of circumstances. You can communicate those criteria across the organization and thus ensure that decisions are being made consistently. You can publish those criteria and use them as a benchmark against which to measure the effectiveness of decisions made in different parts of your organization. You can evolve those criteria in a systematic fashion—testing the effectiveness of decisions and evolving them over time to improve the performance of your business.

And what is the codification of those criteria? By any other name we refer to them as "business rules". Business rules are an independent representation of how the business should behave—the principles and expectations that go into business decisions.

Business rules capture decision criteria in a way that can be applied coherently, comprehensively and consistently across the organization. Further, they enable us to automate the execution of those decisions in our business processes. And by separating the business rules from the technical plumbing of the application we can update automated decision criteria, adjusting those rules as often as new experiences, changes in the environment, or *changes in philosophy* dictate. We can update how our business behaves at the speed of change in our marketplaces.

This book is about Business Rules. It opens by reasoning about the power of separating business rules from the technical infrastructure of our applications. It outlines methods for creating and maintaining business rules. It covers approaches to integrating business rules into our business processes, and for monitoring results and driving improvements to the rules, that in turn, drive improvements in business outcomes. It does so by discussing architectural issues, proposing general solution patterns, and illustrating those patterns for the case of IBM's business rule management system, WebSphere ILOG JRules. Most importantly, it explains how to *manage* rules like you would any other valuable *business asset*.

Quite possibly this will be the most important and comprehensive book you will ever read on the topic of business rules. I highly encourage you to read it from cover to cover and use it to guide your business process and application development activities. Having done so, I'm convinced that you will be in a better position to drive significant improvements to how we leverage Information Technology as a competitive weapon in our business markets.

Rob High, Jr.
IBM Fellow
IBM SOA Foundation, Chief Architect

Foreword II

I first met Jerome and Hafedh at an ILOG event in 2008 when ILOG, then an independent software company focused on business rules, had just donated its work on an Agile Business Rules Development (ABRD) methodology to the open source community. I had heard of this methodology while I was working on Decision Management at FICO, another business rules vendor, but had not had a chance to work with it. I was immediately impressed with both ABRD and the Eclipse Process Framework in which it was presented. I had worked on Ernst & Young's methodology and its automation in the 90s and I understood both the work involved and the value of managing the methodology—not just writing it. ABRD was clearly a well thought out methodology, embodying many best practices for business rules-based development, that could help organizations adopt Decision Management and business rules management systems.

Decision Management is an approach that focusses on automating and improving operational business decisions, including the many micro-decisions that impact a single customer or a single claim. It requires a solid platform for managing decision-making logic—a business rules management system (BRMS)—and a methodology for effectively finding and automating this logic. The combination of business rules and a Decision Management approach results in systems and processes that are simpler, more agile, more aligned with the business and fundamentally smarter. Effective management of the decision logic has improved decision accuracy, compliance and consistency.

Some companies make the mistake of assuming that decision management and business rules can be adopted by an IT department without changing existing governance and development approaches. Others assume that they can handle business rules as part of modeling and managing business processes. In fact, new approaches and techniques are required. Best practices already call for developers to separate the data, user interface and process definitions from applications. Decision Management takes this one step further and separates decision-making logic from the remainder of the technical implementation. Further, it empowers business users and analysts to collaborate effectively with their IT teams and even

to control some of the logic themselves. But extracting decision-making logic as business rules and managing those business rules over time requires new skills, new techniques and new best practices, i.e. a new development methodology.

Among rule development methodologies, ABRD is unique in that it promotes iteration and the early use of a business rules management system. Focussing on incremental and iterative development, it has been specifically developed to handle new artifacts like business rules, decision points and more. It applies the key tenets of the agile manifesto and takes advantage of the power of business rules management systems to deliver on those tenets. Its approach to rule elicitation values "Individuals and interactions over processes and tools". It prototypes early to ensure "Working software over comprehensive documentation". It leverages the ability of non-technical business people to understand and even edit business rules to deliver "Customer collaboration over contract negotiation". Finally, it relies on the faster update and deployment cycles of a business rules management system to ensure projects put "Responding to change over following a plan".

I have helped several companies adopt business rules, using ABRD and the ILOG business rules management system, now part of the IBM WebSphere product suite. These companies have seen tremendous improvements in business agility and in business/IT alignment. Their use of a business rules management system played a big part in these improvements. To be truly successful, however, these companies have also had to adapt and change their approach to systems development and maintenance. Whether they were using agile methods or not for their traditional development, the need for a new approach to effectively apply agile techniques to business rules was clear. ABRD delivered what these companies needed to be successful.

A book on ABRD, then, is both timely and necessary. With this book, Jerome and Hafedh have written more than just a complete guide to ABRD. This book provides an introduction to business rules and to the ABRD methodology. It discusses key ABRD cycles and activities. It outlines key design patterns and covers critical issues in everything from rule authoring to deployment and testing. Rule performance, rule governance and detailed descriptions of how to do all this with IBM's flagship business rules management systems round out a thorough and complete book. If you plan to use business rules to extend and manage the decisions in your operational environment, something I highly recommend, this book will show you how to use an agile approach to do so.

James Taylor

James is CEO and Principal Consultant, Decision Management Solutions and is based in Palo Alto, CA. He is the author, with Neil Raden, of Smart (Enough) Systems (Prentice Hall, 2007) and of numerous chapters on decision management and business rules. He is an active consultant helping companies all over the world implement business rules, and can be reached at james@decisionmanagementsolutions.com.

Preface

Why Business Rules

According to Wordnet, a rule is "a principle or condition that customarily governs behavior" or "a prescribed guide for conduct or action." Businesses, and organizations in general, operate under a number of rules: rules about what services to offer and to whom; rules about how much to charge for those services; rules about how to handle service recipient requests; rules about hiring employees, promoting them, firing them, reimbursing their travel expenses, and paid leave rules; customer relationship management rules; web portal layout rules; salary scales and overtime rules; opening hours rules; emergency behavior guidelines; promotional campaign targeting rules; cross-selling rules, up-selling rules, meeting conduct rules; document disposal recycling and security rules; and so forth. Business rules are everywhere. Every bit of process, task, activity, or function, is governed by rules.

Thus, the question is not *why* business rules, but rather, *how* business rules? Currently, some of the business rules are implicit and thus poorly enforced; those should minimally be written (formalized), if not enforced. Others are written and not enforced. Others yet are poorly written and obscurely enforced. Some are even written and should not – but that is a different story ☺.

The *business rule approach* looks for ways to (1) *write* (elicit, communicate, manage) the important *business rules* in a way that all stakeholders can understand, and (2) *enforce* those business rules *within* the IT infrastructure in a way that supports their traceability and facilitates their maintenance.

The *business rules approach* is no longer the exotic paradigm it was at the turn of the century. Banks are doing it, insurance companies are doing it, phone companies are doing it, retailers are doing it, manufactures are doing it, and government agencies are doing it. This book is not about convincing you of the merits of the business rules approach – it is about helping you adopt it effectively.

Why an *Agile* Business Rule Development Methodology

Business rule pioneers have long recognized that we need a *distinct* development methodology for business rules, one that is different from traditional development methodologies (see justification in Chap. 1). That much we know. But how about *agile*?

Business rules embody *functional requirements*. The business rules approach emphasizes the elicitation, analysis, documentation, and management of such requirements. In fact, rule discovery, discussed in Chap. 4, borrows many techniques from requirements engineering. Thus, "*agile* business rule development" may sound like an oxymoron – how can an approach that puts so much emphasis on requirements be agile?

True. Agility is not a *defining* characteristic of business rule development, except perhaps for the rule maintenance phase, where *IT agility* is achieved through separate authoring and deployment of business rules. To the contrary, most business rule development methodologies put a heavy emphasis on up-front business modeling and analysis. Further, many experts consider business rules within the broader context of enterprise architecture, business–IT alignment, business process reengineering and management, service-oriented everything, or some other intimidating and long-drawn-out are-we-there-yet kind of IT/business transformation that requires deliberate, strategic planning, an unshakeable faith in the outcome, a lot of patience, and deep pockets – in short anything *but* agile.

That is exactly our point. Because agility is not a *given* with business rule development, we need to *engineer* it within business rule development methodologies, and *that is* what *agile business rule development* (ABRD) is about. If we think of a methodology as a pentad of *processes*, *deliverables*, *roles*, *techniques*, and *best practices*, ABRD differs from other business rule methodologies mainly along the *processes* and *best practices* dimensions and, to a lesser extent, on the emphasis (or lack thereof) we put on some of the deliverables. Indeed, ABRD borrows *many* of the business rule–specific techniques and deliverables from other, *rule development* methodologies, including Barbara von Halle's STEP methodology (see von Halle 2002). The *agility* of ABRD, on the other hand, is borrowed from agile methodologies and development principles such as OpenUp, and test-driven development. In particular, ABRD is (1) incremental, (2) iterative, and (3) test-driven. Rather than spending weeks and months discovering and analyzing rules for a complete business function, ABRD puts the emphasis on producing executable, tested – though partial – rulesets since the first few weeks of a project, and *strives* to do that without jeopardizing the quality, perennity, and foresight of the end result.

Our experience in the field shows that ABRD is valuable, feasible, effective, and perfectible! We more than welcome your feedback on the customization and use of ABRD, through personal communication or via the public companion Web site we have set up for the book (http://www.agilebrdevelopment.com) to share comments, criticisms, experiences, information, and insights!

Why This Book

While we think that the ABRD methodology is a story worth telling, it alone does not justify writing – or reading – a book!

Successful adoption of the business rules approach requires four ingredients:

1. *Foundations*, to understand what business rules are (and are not), why you should use the *business rules approach*, and what it can do for you.
2. *Methodology*, to understand *how* to apply the business rules approach, from a *process* point of view, to your business.
3. *Architecture*, to understand how *rule automation*, i.e., how the separate packaging, deployment, and execution of business rules impacts your application.
4. *Implementation*, to actually deliver the technical solution within the context of a particular *business rule management system* (BRMS).

We have long felt that the available business rules literature did not address these four ingredients in an integrated way. There are a number of excellent *foundational* books – most of them are cited in this book – including Ron Ross's *Principles of the Business Rules Approach* (Addison Wesley, 2003) and Tony Morgan's *Business Rules and Information Systems: Aligning IT with Business Goals* (Addison Wesley, 2002). While these books present some business rule–related techniques – some of which are used in this book – they do not provide a step-by-step methodology and do not delve far enough into architecture, let alone implementation. On the methodology front, a number of authors have done a great job, including Barbara von Halle, from whom we gratefully borrow many of the techniques and deliverables of her STEP methodology (see *Business Rules Applied: Building Better Systems Using the Business Rules Approach*, John Wiley & Sons, 2001). However, the book did not (could not) focus on architecture or implementation. James Taylor and Neil Raden's *Smart (Enough) Systems: How to Deliver Competitive Advance by Automating the Hidden Decisions in Your Business* (Prentice-Hall, 2007) focused on how business rules are part of an overall approach to managing and automating decisions but only touched on methodology and the software development life cycle.

From the tool end of the spectrum, we have a number of great books with practical and immediately applicable know-how around specific – typically open-source – rule engines (e.g., JESS) and *budding* business rule management systems (BRMSs, e.g., JBOSS Drools); however, many such books are *definitely* short on methodology (not their focus), short on architecture, and say little about rule management, and governance issues and functionalities.

Hence, the idea of writing this book, which covers all four aspects in significant detail: the *foundations*, in Chaps. 1, 2, and 6; *methodology*, in Chaps. 3, 4, 5, and 16; *architecture and design*, in Chaps. 7, 9, 12, and 14; and *implementation* in Chaps. 8, 10, 11, 13, 15, and 17. We use an insurance case study that deals with *claim processing*. We highlight the major issues in the book text and provide excerpts from the various deliverables. The full versions of the deliverables are available through the companion web portal http://www.agilebrddevelopment.com.

Why JRules

First of all, let us reiterate why we think going to implementation is important. Implementation shows how some design solutions and patterns are *operationalized* within the context of a particular technology. This not only helps the readers to implement the solutions within the chosen technology, but it also helps them in adapting/adopting the solutions to other technologies. The Gang of Four patterns book would not have been the same without the C++ and Smalltalk examples, and that, whether you are implementing in C++, Smalltalk, Java, or C#.

Having decided to go all the way to implementation, we had to pick a business rule management system . . . or two . . . or more. If we were to pick one, it had to be JRules, for several reasons. First of all, it is the one business rule management system (BRMS) that we know best: we have a cumulative experience of 25 years, going through several generations of JRules, and have witnessed major shifts in the industry, in terms of architecture and functionalities. JRules also happens to be a market leader and a mature product, both in terms of deployment architecture and rule management functionality. Our biases notwithstanding, we believe that JRules benefited from great product management, often anticipating and leading market trends.

If we were to pick a second BRMS, which one would it be? Our choice would probably go to JBoss DROOLS, the leading BRMS in open-source tools, both in terms of user community and in terms of entry cost. Including DROOLS would have significantly lengthened this book (another 200 pages) – and the time to write it. And besides, if we pick two, why not pick a third BRMS?

Throughout this book, we strove to identify and separate product/vendor-independent issues, from product-specific features and limitations. This is certainly true for the methodology part, where the contents and semantics of the various work products and deliverables are the same, regardless of the technology. It is also true for rule authoring (a constraint is a constraint, regardless of which BRMS you use), for rule integration (embed rule engines or implement rule execution as a service), for rule testing (unit testing, test scenarios, regression testing, performance tuning, etc.), and for rule governance (rule life cycle, change management, etc.). Out of 18 chapters, only a third (6) are JRules specific.

What happens now *as* JRules evolves? There are three levels of evolution: (1) features, (2) API, and (3) architecture. Features evolve constantly, as menu actions are added here and others are removed from there. That is inevitable, and of no consequence to us: the JRules-specific parts of the book are *not* a product tutorial, anyway; they simply show how to implement some general solution patterns with JRules. As for changes to the API, they seldom break old code. The ones that are *not* related to architecture often consist of limited scope refactorings. With the exception of *Decision Validation Services*, whose *packaging* is fairly recent,[1] the APIs referred to in this book (for ruleset packaging, deployment, execution, performance

[1]By contrast, the core functionality underlying DVS is fairly mature.

tuning, execution server integration, and rule governance) are fairly mature and stable. Changes to the architecture can be more problematic to the shelf life of the material in this book. However, the current architecture uses proven state-of-the-art technologies that are beyond the turbulence of the first years. The portal http://www.agilebrdevelopment.com will maintain information about consequential product updates and will update our operationalization of solution patterns accordingly.

How to Read This Book

This book consists of 18 chapters, organized in eight parts:

Part I, "Introduction," introduces the business rules approach (Chap. 1) and provides example application areas for business rules (Chap. 2).

Part II, "Methodology," focuses on methodology. The *agile business rule development* (ABRD) methodology is presented in Chap. 3. The *rule harvesting cycle* is introduced in Chap. 4, where we talk about rule discovery and analysis, and the *prototyping cycle* (phase) is discussed in Chap. 5.

Part III, "Foundations," covers the basics/main ingredients. Chapter 6 introduces rule engine technology, by going over its history, and explains the inner workings of rule engines, in general, and the JRules rule engine, in particular. Chapter 7 explores the design space for business rule applications and for rule management in the early phases of the rule life cycle. Chapter 8 introduces the JRules BRMS.

Part IV, "Rule Authoring," deals with rule authoring. Chapter 9 explores rule authoring design issues in a technology/vendor-independent way. Chapter 10 discusses JRules artifacts and functionalities for setting up the rule development infrastructure (project structure, business object model) and proposes best practices for it. Chapter 11 discusses rule authoring per se, where we introduce the JRules rule languages and artifacts, and rule execution orchestration.

Part V, "Rule Deployment," deals with ruleset deployment and execution; Chap. 12 discusses deployment and execution issues, in general, whereas Chap. 13 explores deployment and execution options in JRules.

Part VI, "Rule Testing," deals with testing. Chapter 14 discusses rule testing and validation issues, in general, whereas Chap. 15 explores JRules functionality for rule testing, tracing, and performance monitoring.

Part VII, "Rule Governance," deals with rule governance. Chapter 16 introduces rule governance and discusses the main process and design issues. Chapter 17 explores JRules support for rule governance.

Part VIII, "Epilogue," concludes this book with a short epilogue.

Clearly, by choosing to address *foundations*, *methodology*, *architecture*, and *implementation*, this book caters to five different audiences:

- *Project managers* will find a pragmatic, proven methodology for delivering and maintaining business rule applications.

- *Business analysts* will find a methodology that they can use for rule discovery and analysis, and a number of guidelines and best practices for rule authoring, and for structuring rules during development.
- *Rule authors* will find a number of guidelines and best practices for rule authoring, in general, and detailed explanations about rule artifacts and rule authoring languages in JRules.
- *Application and software architects* will find an exploration of the design space for business rule applications, and a number or proven architectural and design patterns, in general, and for the case of JRules.
- *Developers* will find practical design and coding guidelines for implementing design choices, in general, and using JRules.

Incidentally, CTOs and product/business line managers will also find some value in this book; thanks to our explanation of the business rules approach, to the example application areas, and to a discussion of rule governance issues, but they are probably better off with other foundational books such as those mentioned earlier.

The following table shows reading paths for the different audiences:

Target audience	Should-read chapters/parts	Optional chapters
Project manager	Parts I and II: Chaps. 1–5, Chaps. 8, 16, and 18	Chaps. 7 and 14
Application architect	Parts I and II: Chaps. 1–5, Chaps. 8, 12, 14, 16, and 18	
Software architect	Part I: Chaps. 1–2, Chap. 3; Part III: Chaps. 6–8; Part V: Chaps. 12–13; Part VI: Chaps. 14–15; Part VII: Chaps. 16–17); and Part VIII: Chap. 18	Chaps. 4 and 5
Business analyst	Part I: Chaps. 1–2, Chaps. 3, 4, 8, 9, 14, 16, and 18	Chap. 7
Rule author	Part I: Chaps. 1–2, Chaps. 3, 8; Part IV: Chaps. 9–11, Chap. 16; Part VIII: Chap. 18	
Developer	Part I: Chaps. 1–2, Chaps. 3 and 5; Part III: Chaps. 6–8, Chap. 10; Part V: Chaps. 12–13; Part VI: Chaps. 14–15; Part VII: Chaps. 16–17; and Part VIII: Chap. 18	

Acknowledgments

This book has been an on-and-off project for many years. Vilas Tulachan, an independent J2EE consultant and author, and a JRules consultant and trainer, has revived an earlier incarnation of this book project, which, while it did not materialize in its earlier form, kept us talking about it above the noise level, until a concrete book proposal was submitted to Ralf Gerstner, our indefatigable Springer editor, in the fall of 2007.

We wish to thank Ralf for his legendary patience with us through many (self-imposed) missed time targets. Thanks to ABRD, we are much better at delivering business rule solutions than we have been at delivering this book!

ABRD is the open-source descendant of the proprietary ILOG ISIS (ILOG Solution Implementation Standard) methodology. Our thanks to the members of the ISIS team, namely, Pierre Berlandier, who has written extensively about rule governance, and Jean Pommier, who supported the development of ABRD, its open publication – and the writing of this book!

Our sincerest thanks go to Tonya Teyssier, a conscientious, patient, and generous JRules curriculum developer from IBM WebSphere Education, who sacrificed many evenings and weekends to help us write – and think – clearly the first chapters of the book. She has become a master of euphemisms in "constructively criticizing" some of the earlier drafts.

Eric Charpentier, a JRules consultant extraordinaire, who excels at everything he does, provided us with very valuable and timely feedback on *all* the chapters of the book. He certainly helped us a great deal in improving the organization and pedagogy of many chapters of the book. Eric blogs about topics ranging from scorecards to rule governance (see http://www.primatek.ca/blog).

James Taylor, a leading authority on decision management, including business rules, and analytics, and an independent consultant, speaker, and author, volunteered to read a complete draft of the book, and provided us with valuable, timely, concise, to the point (and witty) feedback, James-style! He blogs extensively about decision management (check JT on EDM, at http://jtonedm.com/), and has authored, with Neil Raden, *Smart Enough Systems: How to Deliver Competitive*

Advantage by Automating Hidden Decisions (Prentice-Hall, 2007), which is becoming a classic on decision management.

We both wish to thank our respective families who, like families of all authors, have to put up with absentee – or absent-minded – father/partner for a never-ending book project. Are we there yet? Yes, we are ... till the next book ☺.

December 2010 Hafedh Mili and Jérôme Boyer

Contents

Part II Methodology