

Implementing Domain Specific Languages With Xtext And Xtend By Bettini Lorenzo 2013 Paperback

Dijkstra once wrote that computer science is no more about computers than astronomy is about telescopes. Despite the many incredible advances in computer science from times that predate practical mechanical computing, there is still a myriad of fundamental questions in understanding the interface between computers and the rest of the world. Why is it still hard to mechanize many tasks that seem to be fundamentally routine, even as we see ever-increasing capacity for raw mechanical computing? The disciplined study of domain-specific languages (DSLs) is an emerging area in computer science, and is one which has the potential to revolutionize the field, and bring us closer to answering this question. DSLs are formalisms that have four general characteristics. – They relate to a well-defined domain of discourse, be it controlling traffic lights or space ships. – They have well-defined notation, such as the ones that exist for prescribing music, dance routines, or strategy in a football game. – The informal or intuitive meaning of the notation is clear. This can easily be overlooked, especially since intuitive meaning can be expressed by many different notations that may be received very

differently by users. – The formal meaning is clear and mechanizable, as is, hopefully, the case for the instructions we give to our bank or to a merchant online.

Learn how to manipulate functions and expressions to modify how the R language interprets itself. This book is an introduction to metaprogramming in the R language, so you will write programs to manipulate other programs. Metaprogramming in R shows you how to treat code as data that you can generate, analyze, or modify. R is a very high-level language where all operations are functions and all functions are data that can be manipulated. This book shows you how to leverage R's natural flexibility in how function calls and expressions are evaluated, to create small domain-specific languages to extend R within the R language itself. What You'll Learn Find out about the anatomy of a function in R Look inside a function call Work with R expressions and environments Manipulate expressions in R Use substitutions Who This Book Is For Those with at least some experience with R and certainly for those with experience in other programming languages. The development of modern complex software-intensive systems often involves the use of multiple DSMLs that capture different system aspects. Supporting coordinated use of DSMLs leads to what we call the globalization of modeling languages, that is, the use of multiple modeling languages to support

coordinated development of diverse aspects of a system. In this book, a number of articles describe the vision and the way globalized DSMLs currently assist integrated DSML support teams working on systems that span many domains and concerns to determine how their work on a particular aspect influences work on other aspects. Globalized DSMLs offer support for communicating relevant information, and for coordinating development activities and associated technologies within and across teams, in addition to providing support for imposing control over development artifacts produced by multiple teams. DSMLs can be used to support socio-technical coordination by providing the means for stakeholders to bridge the gap between how they perceive a problem and its solution, and the programming technologies used to implement a solution. They also support coordination of work across multiple teams. DSMLs developed in an independent manner to meet the specific needs of domain experts have an associated framework that regulates interactions needed to support collaboration and work coordination across different system domains. The articles in the book describe how multiple heterogeneous modeling languages (or DSMLs) can be related to determine how different aspects of a system influence each other. The book includes a research roadmap that broadens the current DSML research focus beyond the

development of independent DSMLs to one that provides support for globalized DSMLs.

UKSC 84 contains the proceedings of the 1984 United Kingdom Simulation Council Conference on Computer Simulation held at the University of Bath, England. The papers describe computer simulation techniques and their applications and cover topics ranging from simulation methodology and software to the various applications of computer simulation in areas such as policy decision-making and planning, biology and medicine, and education. This book is comprised of 52 chapters divided into nine sections and begins by describing an advanced continuous-system simulation language called ESL (ESA Simulation Language), an initiative of the European Space Agency. The papers that follow explore other simulation software, such as MANIP, SYSMOD, COSMOS, Ada, SDL (Simulation Development Language), and SPIRO (Suite of Programs for the Investigation of Recondite Objects). The discussion then turns to a methodology based on artificial intelligence for the design and development of large-scale computer simulations; a formalism for specifying continuous or fixed time-step simulation models that is a straightforward extension of the block-oriented languages, with emphasis on superblocks and tessellations; and simulation of manufacturing and control systems. This book concludes with a chapter that describes a highly

efficient compactor for a radar digital database. This monograph will be of interest to students and professionals working in the field of computer simulation.

This Festschrift volume has been published in honor of Frank de Boer, on the occasion of his 60th birthday. Frank S. de Boer is a prominent member of the research community in formal methods and theoretical computer science. A brief look at his lengthy publication list reveals a broad area of interest and a versatile *modus operandi* with: logic and constraint programming; deductive proof systems, soundness, and completeness; semantics, compositionality, and full abstraction; process algebra and decidability; multithreading and actor-based concurrency; agent programming, ontologies, and modal logic; real-time systems, timed automata, and schedulability; enterprise architectures, choreography, and coordination; testing and runtime monitoring; and cloud computing and service-level agreements. For a while, he also liked failures, especially in semantics, and optimistically concluded with the failure of failures. In fact, Frank has an opportunistic approach to research. Rather than seeing obstacles, he finds opportunities.

Get up to speed on Scala, the JVM language that offers all the benefits of a modern object model, functional programming, and an advanced type system. Packed with code examples, this

Bookmark File PDF Implementing Domain Specific Languages With Xtext And Xtend By Bettini Lorenzo 2013 Paperback

comprehensive book shows you how to be productive with the language and ecosystem right away, and explains why Scala is ideal for today's highly scalable, data-centric applications that support concurrency and distribution. This second edition covers recent language features, with new chapters on pattern matching, comprehensions, and advanced functional programming. You'll also learn about Scala's command-line tools, third-party tools, libraries, and language-aware plugins for editors and IDEs. This book is ideal for beginning and advanced Scala developers alike. Program faster with Scala's succinct and flexible syntax Dive into basic and advanced functional programming (FP) techniques Build killer big-data apps, using Scala's functional combinators Use traits for mixin composition and pattern matching for data extraction Learn the sophisticated type system that combines FP and object-oriented programming concepts Explore Scala-specific concurrency tools, including Akka Understand how to develop rich domain-specific languages Learn good design techniques for building scalable and robust Scala applications

This book is a practical tutorial, walking the reader through examples of building DSLs with Groovy covering meta-programming with Groovy. Some complex concepts are covered in the book but we go through these in a clear and methodically way so that readers will gain a good working knowledge of

Bookmark File PDF Implementing Domain Specific Languages With Xtext And Xtend By Bettini Lorenzo 2013 Paperback

the techniques involved. This book is for Java software developers who have an interest in building domain scripting into their Java applications. No knowledge of Groovy is required, although it will be helpful. The book does not teach Groovy, but quickly introduces the basic ideas of Groovy. An experienced Java developer should have no problems with these and move quickly on to the more involved aspects of creating DSLs with Groovy. No experience of creating a DSL is required. The book should also be useful for experienced Groovy developers who have so far only used Groovy DSLs such as Groovy builders and would like to start building their own Groovy-based DSLs.

"This book presents current research on all aspects of domain-specific language for scholars and practitioners in the software engineering fields, providing new results and answers to open problems in DSL research"--

Model-Driven Software Development (MDS) is currently a highly regarded development paradigm among developers and researchers. With the advent of OMG's MDA and Microsoft's Software Factories, the MDS approach has moved to the centre of the programmer's attention, becoming the focus of conferences such as OOPSLA, JAOS and OOP. MDS is about using domain-specific languages to create models that express application structure or behaviour in an efficient and domain-specific way. These models are subsequently transformed into executable code by a sequence of model transformations. This practical guide for

Bookmark File PDF Implementing Domain Specific Languages With Xtext And Xtend By Bettini Lorenzo 2013 Paperback

software architects and developers is peppered with practical examples and extensive case studies. International experts deliver:

- * A comprehensive overview of MDS and how it relates to industry standards such as MDA and Software Factories.
- * Technical details on meta modeling, DSL construction, model-to-model and model-to-code transformations, and software architecture.
- * Invaluable insight into the software development process, plus engineering issues such as versioning, testing and product line engineering.
- * Essential management knowledge covering economic and organizational topics, from a global perspective.

Get started and benefit from some practical support along the way!

Build domain specific languages (DSLs) using Java's most popular functional programming language: Scala. This book introduces the basics of Scala and DSLs using a series of practical examples. In *Practical Scala DSLs*, you'll learn to create pragmatic and complete code examples that explain the actual use of DSLs with Scala: a web API and microservices; a custom language; a mobile app; a Forex system; a game; and cloud applications. At the end of this unique book, you'll be able to describe the differences between external and internal DSLs; understand when and how to apply DSLs; create DSLs using Scala; and even create a DSL using another programming language.

You'll Learn

- Build DSLs in Scala
- Write a web API and microservices
- Create a custom language
- Apply DSLs to mobile apps development, a Forex trading system, game development, and more
- Discover the role of DSLs in cloud development
- Integrate DSLs as part of a DevOps program or structure
- Build internal and external DSLs

Who This Book Is For

Experienced Java coders with at least some prior experience with Scala. You may be new to DSLs.

Describes ways to incorporate domain modeling into software

Bookmark File PDF Implementing Domain Specific Languages With Xtext And Xtend By Bettini Lorenzo 2013 Paperback development.

When carefully selected and used, Domain-Specific Languages (DSLs) may simplify complex code, promote effective communication with customers, improve productivity, and unclog development bottlenecks. In *Domain-Specific Languages*, noted software development expert Martin Fowler first provides the information software professionals need to decide if and when to utilize DSLs. Then, where DSLs prove suitable, Fowler presents effective techniques for building them, and guides software engineers in choosing the right approaches for their applications. This book's techniques may be utilized with most modern object-oriented languages; the author provides numerous examples in Java and C#, as well as selected examples in Ruby. Wherever possible, chapters are organized to be self-standing, and most reference topics are presented in a familiar patterns format. Armed with this wide-ranging book, developers will have the knowledge they need to make important decisions about DSLs—and, where appropriate, gain the significant technical and business benefits they offer. The topics covered include: How DSLs compare to frameworks and libraries, and when those alternatives are sufficient Using parsers and parser generators, and parsing external DSLs Understanding, comparing, and choosing DSL language constructs Determining whether to use code generation, and comparing code generation strategies Previewing new language workbench tools for creating DSLs

This book constitutes the thoroughly refereed post-conference proceedings of the Second International Conference on Software Language Engineering, SLE 2009, held in Denver, CO, USA, in October 2009. The 15 revised full papers and 6 revised short paper presented together with 2 tool demonstration papers were carefully reviewed and selected from 75 initial submissions. The papers are

Bookmark File PDF Implementing Domain Specific Languages With Xtext And Xtend By Bettini Lorenzo 2013 Paperback

organized in topical sections on language and model evolution, variability and product lines, parsing, compilation, and demo, modularity in languages, and metamodeling and demo.

This book constitutes the thoroughly refereed post-conference proceedings of the Second International Conference on Runtime Verification, RV 2011, held in San Francisco, USA, in September 2011. The 24 revised full papers presented together with 3 invited papers, 4 tutorials and 4 tool demonstrations were carefully reviewed and selected from 71 submissions. The papers are organized in topical sections on parallelism and deadlocks, malware detection, temporal constraints and concurrency bugs, sampling and specification conformance, real-time, software and hardware systems, memory transactions, tools; foundational techniques and multi-valued approaches.

“For software developers of all experience levels looking to improve their results, and design and implement domain-driven enterprise applications consistently with the best current state of professional practice, *Implementing Domain-Driven Design* will impart a treasure trove of knowledge hard won within the DDD and enterprise application architecture communities over the last couple decades.” –Randy Stafford, Architect At-Large, Oracle Coherence Product Development

“This book is a must-read for anybody looking to put DDD into practice.” –Udi Dahan, Founder of NServiceBus

Implementing Domain-Driven Design presents a top-down approach to understanding domain-driven design (DDD) in a way that fluently connects strategic patterns to fundamental tactical programming tools. Vaughn Vernon couples guided approaches to implementation with modern architectures, highlighting the importance and value of focusing on the business domain while balancing technical considerations. Building on Eric Evans’ seminal book, *Domain-Driven*

Bookmark File PDF Implementing Domain Specific Languages With Xtext And Xtend By Bettini Lorenzo 2013 Paperback

Design, the author presents practical DDD techniques through examples from familiar domains. Each principle is backed up by realistic Java examples—all applicable to C# developers—and all content is tied together by a single case study: the delivery of a large-scale Scrum-based SaaS system for a multitenant environment. The author takes you far beyond “DDD-lite” approaches that embrace DDD solely as a technical toolset, and shows you how to fully leverage DDD’s “strategic design patterns” using Bounded Context, Context Maps, and the Ubiquitous Language. Using these techniques and examples, you can reduce time to market and improve quality, as you build software that is more flexible, more scalable, and more tightly aligned to business goals. Coverage includes Getting started the right way with DDD, so you can rapidly gain value from it Using DDD within diverse architectures, including Hexagonal, SOA, REST, CQRS, Event-Driven, and Fabric/Grid-Based Appropriately designing and applying Entities—and learning when to use Value Objects instead Mastering DDD’s powerful new Domain Events technique Designing Repositories for ORM, NoSQL, and other databases

The second instance of the international summer school on Generative and Transformational Techniques in Software Engineering (GTTSE 2007) was held in Braga, Portugal, during July 2–7, 2007. This volume contains an augmented selection of the material presented at the school, including full tutorials, short tutorials, and contributions to the participants workshop. The GTTSE summer school series brings together PhD students, lecturers, technology presenters, as well as other researchers and practitioners who are interested in the generation and the transformation of programs, data, models, metamodels, documentation, and entire software systems. This concerns many areas of software engineering: software reverse and re-engineering, model-driven engineering,

Bookmark File PDF Implementing Domain Specific Languages With Xtext And Xtend By Bettini Lorenzo 2013 Paperback

automated software engineering, generic language technology, to name a few. These areas differ with regard to the specific sorts of metamodels (or grammars, schemas, formats etc.) that underlie the involved artifacts, and with regard to the specific techniques that are employed for the generation and the transformation of the artifacts. The first instance of the school was held in 2005 and its proceedings appeared as volume 4143 in the LNCS series.

Gain an accelerated introduction to domain-specific languages in R, including coverage of regular expressions. This compact, in-depth book shows you how DSLs are programming languages specialized for a particular purpose, as opposed to general purpose programming languages. Along the way, you'll learn to specify tasks you want to do in a precise way and achieve programming goals within a domain-specific context. Domain-Specific Languages in R includes examples of DSLs including large data sets or matrix multiplication; pattern matching DSLs for application in computer vision; and DSLs for continuous time Markov chains and their applications in data science. After reading and using this book, you'll understand how to write DSLs in R and have skills you can extrapolate to other programming languages.

What You'll Learn
Program with domain-specific languages using R
Discover the components of DSLs
Carry out large matrix expressions and multiplications
Implement metaprogramming with DSLs
Parse and manipulate expressions

Who This Book Is For
Those with prior programming experience. R knowledge is helpful but not required.

Generic Tools, Specific Languages (GTSL) is an

Bookmark File PDF Implementing Domain Specific Languages With Xtext And Xtend By Bettini Lorenzo 2013 Paperback

approach for developing tools and applications in a way that supports easier and more meaningful adaptation to specific domains. To achieve this goal, GTSL generalizes programming language IDEs to domains traditionally not addressed by languages and IDEs. At its core, GTSL represents applications as documents/programs/models expressed with suitable languages. Application functionality is provided through an IDE that is aware of the languages and their semantics. The IDE provides editing support, and also directly integrates domain-specific analyses and execution services. Applications and their languages can be adapted to increasingly specific domains using language engineering; this includes developing incremental extensions to existing languages or creating additional, tightly integrated languages. Language workbenches act as the foundation on which such applications are built. mbeddr is an extensible set of integrated languages for embedded software development built using the Generic Tools, Specific Languages approach.

Expert F# 2.0 is about practical programming in a beautiful language that puts the power and elegance of functional programming into the hands of professional developers. In combination with .NET, F# achieves unrivaled levels of programmer productivity and program clarity. Expert F# 2.0 is The authoritative guide to F# by the inventor of F# A comprehensive reference of F# concepts, syntax, and features A treasury of expert F# techniques for practical, real-world programming F# isn't just another functional programming language. It's a

Bookmark File PDF Implementing Domain Specific Languages With Xtext And Xtend By Bettini Lorenzo 2013 Paperback

general-purpose language ideal for real-world development. F# seamlessly integrates functional, imperative, and object-oriented programming styles so you can flexibly and elegantly solve any programming problem. Whatever your background, you'll find that F# is easy to learn, fun to use, and extraordinarily powerful. F# will change the way you think about—and go about—programming. Written by F#'s inventor and two major contributors to its development, Expert F# 2.0 is the authoritative, comprehensive, and in-depth guide to the language and its use. Designed to help others become experts, the first part of the book quickly yet carefully describes the F# language. The second part then shows how to use F# elegantly for a wide variety of practical programming tasks. The world's foremost experts in F# show you how to program in F# the way they do!

Your success—and sanity—are closer at hand when you work at a higher level of abstraction, allowing your attention to be on the business problem rather than the details of the programming platform. Domain Specific Languages—"little languages" implemented on top of conventional programming languages—give you a way to do this because they model the domain of your business problem. DSLs in Action introduces the concepts and definitions a developer needs to build high-quality domain specific languages. It provides a solid foundation to the usage as well as implementation aspects of a DSL, focusing on the necessity of applications speaking the language of the domain. After reading this book, a programmer will be able to design APIs that make better

Bookmark File PDF Implementing Domain Specific Languages With Xtext And Xtend By Bettini Lorenzo 2013 Paperback

domain models. For experienced developers, the book addresses the intricacies of domain language design without the pain of writing parsers by hand. The book discusses DSL usage and implementations in the real world based on a suite of JVM languages like Java, Ruby, Scala, and Groovy. It contains code snippets that implement real world DSL designs and discusses the pros and cons of each implementation. Purchase of the print book comes with an offer of a free PDF, ePub, and Kindle eBook from Manning. Also available is all code from the book.

What's Inside

- Tested, real-world examples
- How to find the right level of abstraction
- Using language features to build internal DSLs
- Designing parser/combinator-based little languages

Software practitioners are rapidly discovering the immense value of Domain-Specific Languages (DSLs) in solving problems within clearly definable problem domains. Developers are applying DSLs to improve productivity and quality in a wide range of areas, such as finance, combat simulation, macro scripting, image generation, and more. But until now, there have been few practical resources that explain how DSLs work and how to construct them for optimal use. Software Language Engineering fills that need. Written by expert DSL consultant Anneke Kleppe, this is the first comprehensive guide to successful DSL design. Kleppe systematically introduces and explains every ingredient of an effective language specification, including its description of concepts, how those concepts are denoted, and what those concepts mean in relation to the problem domain. Kleppe carefully illuminates good

Bookmark File PDF Implementing Domain Specific Languages With Xtext And Xtend By Bettini Lorenzo 2013 Paperback

design strategy, showing how to maximize the flexibility of the languages you create. She also demonstrates powerful techniques for creating new DSLs that cooperate well with general-purpose languages and leverage their power. Completely tool-independent, this book can serve as the primary resource for readers using Microsoft DSL tools, the Eclipse Modeling Framework, openArchitectureWare, or any other DSL toolset. It contains multiple examples, an illustrative running case study, and insights and background information drawn from Kleppe's leading-edge work as a DSL researcher. Specific topics covered include

- Discovering the types of problems that DSLs can solve, and when to use them
- Comparing DSLs with general-purpose languages, frameworks, APIs, and other approaches
- Understanding the roles and tools available to language users and engineers
- Creating each component of a DSL specification
- Modeling both concrete and abstract syntax
- Understanding and describing language semantics
- Defining textual and visual languages based on object-oriented metamodeling and graph transformations
- Using metamodels and associated tools to generate grammars
- Integrating object-oriented modeling with graph theory
- Building code generators for new languages
- Supporting multilanguage models and programs

This book provides software engineers with all the guidance they need to create DSLs that solve real problems more rapidly, and with higher-quality code.

Creating your own domain-specific languages (DSLs) is both challenging and exhilarating. DSLs give users a

Bookmark File PDF Implementing Domain Specific Languages With Xtext And Xtend By Bettini, Lorenzo 2013 Paperback

way to interact with your applications more effectively, and Kotlin is a fantastic language to serve as a host for internal DSLs, because it greatly reduces the pain and effort of design and development. But implementing DSLs on top of Kotlin requires understanding the key strengths of the language and knowing how to apply them appropriately. Learn to avoid the pitfalls and leverage the language while creating your own elegant, fluent, concise, and robust DSLs using Kotlin. Internal DSLs remove the burdens of implementing a full blown language compiler. The host language quickly becomes your ally to creating DSLs, but the syntax you can choose for your DSLs is limited to what the host language allows. You can work around the limitations by tactfully bending the rules and exploiting the language capabilities. Learn the power of Kotlin and ways to design with it, in the context of crafting internal DSLs

Start by learning ways to exploit the flexibilities of Kotlin to make your DSLs fluent, expressive, and concise. Then pick up techniques to extend the language with domain specific properties and functions. Quickly move ahead to tie your DSL snippets into the runtime environment and context of execution of your applications. Design to prevent any non-sensical syntax in your DSL that may otherwise be valid in the host language. Finally, learn techniques to gracefully handle errors. Practice using the multiple examples that are included in each chapter. Fire up your editor and follow along each example to become proficient in designing and implementing your own internal DSLs using Kotlin. What You Need: Kotlin version 1.3 or later and your favorite Kotlin IDE or code

Bookmark File PDF Implementing Domain Specific Languages With Xtext And Xtend By Bettini Lorenzo 2013 Paperback editor.

The definitive resource on domain-specific languages: based on years of real-world experience, relying on modern language workbenches and full of examples. Domain-Specific Languages are programming languages specialized for a particular application domain. By incorporating knowledge about that domain, DSLs can lead to more concise and more analyzable programs, better code quality and increased development speed. This book provides a thorough introduction to DSL, relying on today's state of the art language workbenches. The book has four parts: introduction, DSL design, DSL implementation as well as the role of DSLs in various aspects of software engineering. Part I Introduction: This part introduces DSLs in general and discusses their advantages and drawbacks. It also defines important terms and concepts and introduces the case studies used in the most of the remainder of the book. Part II DSL Design: This part discusses the design of DSLs - independent of implementation techniques. It reviews seven design dimensions, explains a number of reusable language paradigms and points out a number of process-related issues. Part III DSL Implementation: This part provides details about the implementation of DSLs with lots of code. It uses three state-of-the-art but quite different language workbenches: JetBrains MPS, Eclipse Xtext and TU Delft's Spoofox. Part IV DSLs and Software Engineering: This part discusses the use of DSLs for requirements, architecture, implementation and product line engineering, as well as their roles as a developer utility and for implementing business logic.

Bookmark File PDF Implementing Domain Specific Languages With Xtext And Xtend By Bettini Lorenzo 2013 Paperback

The book is available as a printed version (the one you are looking at) and as a PDF. For details see the book's companion website at <http://dslbook.org>

A step-by-step guide that enables you to quickly implement a DSL with Xtext and Xtend in a test-driven way with the aid of simplified examples. This book is for programmers who want to learn about Xtext and how to use it to implement a DSL (or a programming language) together with Eclipse IDE tooling. It assumes that the user is familiar with Eclipse and its functionality. Existing basic knowledge of a compiler implementation would be useful, though not strictly required, since the book will explain all the stages of the development of a DSL. Achieve Breakthrough Productivity and Quality with MDD and Eclipse-Based DSLs Domain-specific languages (DSLs) and model-driven development (MDD) offer software engineers powerful new ways to improve productivity, enhance quality, and insulate systems from rapid technological change. Now, there's a pragmatic, start-to-finish guide to creating DSLs and using MDD techniques with the powerful open source Eclipse platform. In Eclipse Modeling Project, Richard C. Gronback illuminates both the principles and techniques software professionals need to master, offering insights that will be invaluable to developers working with any tool or platform. As coleader of the Eclipse Modeling Project, Gronback is singularly well-positioned to demonstrate DSLs and MDD at work in Eclipse. Gronback systematically introduces each of the Eclipse technologies that can be used in DSL and MDD development. Throughout, he introduces key concepts

Bookmark File PDF Implementing Domain Specific Languages With Xtext And Xtend By Bettini Lorenzo 2013 Paperback

and technologies in the context of a complete worked example and presents new best practices and never-before published techniques. He also covers Eclipse projects discussed in no other book, including Query/View/Transformation (QVT) and the Graphical Modeling Framework (GMF)—a project the author personally leads. Eclipse Modeling Project gives software practitioners all the knowledge they need to explore the remarkable potential of DSLs and MDD—and includes coverage of Why a model-based approach enables the rapid customization of high-quality solutions within the product line paradigm How the Eclipse Modeling Project’s capabilities can be used to efficiently create new DSLs Powerful techniques for developing DSL abstract syntax, graphical notation, and textual syntax How to build Model-to-Model (M2M) and Model-to-Text (M2T) transformations—including a powerful new M2M implementation of the Object Management Group’s QVT Operational Mapping Language (OML) Efficiently packaging and deploying DSLs with Eclipse Complete reference sections for the Graphical Editing Framework (GEF), GMF runtime and tooling, QVT OML, Xpand, and more

Written for developers who need to create user-facing DSLs, *Domain-Specific Languages Made Easy* unlocks clear and practical methods to create DSLs with easy-to-use interfaces. Imagine if your non-technical clients could safely produce software without the need for anyone to manually write code. Domain-specific languages are purpose-built programming interfaces that make that possible—no programming experience

Bookmark File PDF Implementing Domain Specific Languages With Xtext And Xtend By Bettini Lorenzo 2013 Paperback

required. Written for developers who need to create user-facing DSLs, *Domain-Specific Languages Made Easy* unlocks clear and practical methods to create DSLs with easy-to-use interfaces. Author Meinte Boersma lays out an iterative process for creating languages accessible to domain experts such as operations specialists, data analysts, and financial experts. You'll start with an overview of software language engineering before diving into the unique projectional editing paradigm that makes it easy to produce DSLs for business. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications.

Programmers run into parsing problems all the time. Whether it's a data format like JSON, a network protocol like SMTP, a server configuration file for Apache, a PostScript/PDF file, or a simple spreadsheet macro language--ANTLR v4 and this book will demystify the process. ANTLR v4 has been rewritten from scratch to make it easier than ever to build parsers and the language applications built on top. This completely rewritten new edition of the bestselling *Definitive ANTLR Reference* shows you how to take advantage of these new features. Build your own languages with ANTLR v4, using ANTLR's new advanced parsing technology. In this book, you'll learn how ANTLR automatically builds a data structure representing the input (parse tree) and generates code that can walk the tree (visitor). You can use that combination to implement data readers, language interpreters, and translators. You'll start by learning how to identify grammar patterns in language reference manuals and then slowly start building

Bookmark File PDF Implementing Domain Specific Languages With Xtext And Xtend By Bettini, Lorenzo 2013 Paperback

increasingly complex grammars. Next, you'll build applications based upon those grammars by walking the automatically generated parse trees. Then you'll tackle some nasty language problems by parsing files containing more than one language (such as XML, Java, and Javadoc). You'll also see how to take absolute control over parsing by embedding Java actions into the grammar. You'll learn directly from well-known parsing expert Terence Parr, the ANTLR creator and project lead. You'll master ANTLR grammar construction and learn how to build language tools using the built-in parse tree visitor mechanism. The book teaches using real-world examples and shows you how to use ANTLR to build such things as a data file reader, a JSON to XML translator, an R parser, and a Java class->interface extractor. This book is your ticket to becoming a parsing guru! What You Need: ANTLR 4.0 and above. Java development tools. Ant build system optional(needed for building ANTLR from source)

Learn to build configuration file readers, data readers, model-driven code generators, source-to-source translators, source analyzers, and interpreters. You don't need a background in computer science--ANTLR creator Terence Parr demystifies language implementation by breaking it down into the most common design patterns. Pattern by pattern, you'll learn the key skills you need to implement your own computer languages. Knowing how to create domain-specific languages (DSLs) can give you a huge productivity boost. Instead of writing code in a general-purpose programming language, you can first build a custom language tailored to make you efficient in

Bookmark File PDF Implementing Domain Specific Languages With Xtext And Xtend By Bettini Lorenzo 2013 Paperback

a particular domain. The key is understanding the common patterns found across language implementations. Language Design Patterns identifies and condenses the most common design patterns, providing sample implementations of each. The pattern implementations use Java, but the patterns themselves are completely general. Some of the implementations use the well-known ANTLR parser generator, so readers will find this book an excellent source of ANTLR examples as well. But this book will benefit anyone interested in implementing languages, regardless of their tool of choice. Other language implementation books focus on compilers, which you rarely need in your daily life. Instead, Language Design Patterns shows you patterns you can use for all kinds of language applications. You'll learn to create configuration file readers, data readers, model-driven code generators, source-to-source translators, source analyzers, and interpreters. Each chapter groups related design patterns and, in each pattern, you'll get hands-on experience by building a complete sample implementation. By the time you finish the book, you'll know how to solve most common language implementation problems.

Learn how to implement a DSL with Xtext and Xtend using easy-to-understand examples and best practices

About This Book Leverage the latest features of Xtext and Xtend to develop a domain-specific language.

Integrate Xtext with popular third party IDEs and get the best out of both worlds. Discover how to test a DSL implementation and how to customize runtime and IDE aspects of the DSL

Who This Book Is For This book is

Bookmark File PDF Implementing Domain Specific Languages With Xtext And Xtend By Bettini Lorenzo 2013 Paperback

targeted at programmers and developers who want to create a domain-specific language with Xtext. They should have a basic familiarity with Eclipse and its functionality. Previous experience with compiler implementation can be helpful but is not necessary since this book will explain all the development stages of a DSL. What You Will Learn Write Xtext grammar for a DSL; Use Xtend as an alternative to Java to write cleaner, easier-to-read, and more maintainable code; Build your Xtext DSLs easily with Maven/Tycho and Gradle; Write a code generator and an interpreter for a DSL; Explore the Xtext scoping mechanism for symbol resolution; Test most aspects of the DSL implementation with JUnit; Understand best practices in DSL implementations with Xtext and Xtend; Develop your Xtext DSLs using Continuous Integration mechanisms; Use an Xtext editor in a web application In Detail Xtext is an open source Eclipse framework for implementing domain-specific languages together with IDE functionalities. It lets you implement languages really quickly; most of all, it covers all aspects of a complete language infrastructure, including the parser, code generator, interpreter, and more. This book will enable you to implement Domain Specific Languages (DSL) efficiently, together with their IDE tooling, with Xtext and Xtend. Opening with brief coverage of Xtext features involved in DSL implementation, including integration in an IDE, the book will then introduce you to Xtend as this language will be used in all the examples throughout the book. You will then explore the typical programming development workflow with Xtext when we modify the

Bookmark File PDF Implementing Domain Specific Languages With Xtext And Xtend By Bettini Lorenzo 2013 Paperback

grammar of the DSL. Further, the Xtend programming language (a fully-featured Java-like language tightly integrated with Java) will be introduced. We then explain the main concepts of Xtext, such as validation, code generation, and customizations of runtime and UI aspects. You will have learned how to test a DSL implemented in Xtext with JUnit and will progress to advanced concepts such as type checking and scoping. You will then integrate the typical Continuous Integration systems built in to Xtext DSLs and familiarize yourself with Xbase. By the end of the book, you will manually maintain the EMF model for an Xtext DSL and will see how an Xtext DSL can also be used in IntelliJ. Style and approach A step-by-step-tutorial with illustrative examples that will let you master using Xtext and implementing DSLs with its custom language, Xtend.

Practical Software Architecture Solutions from the Legendary Robert C. Martin ("Uncle Bob") By applying universal rules of software architecture, you can dramatically improve developer productivity throughout the life of any software system. Now, building upon the success of his best-selling books Clean Code and The Clean Coder, legendary software craftsman Robert C. Martin ("Uncle Bob") reveals those rules and helps you apply them. Martin's Clean Architecture doesn't merely present options. Drawing on over a half-century of experience in software environments of every imaginable type, Martin tells you what choices to make and why they are critical to your success. As you've come to expect from Uncle Bob, this book is packed with direct, no-nonsense solutions for the real challenges you'll

Bookmark File PDF Implementing Domain Specific Languages With Xtext And Xtend By Bettini Lorenzo 2013 Paperback

face—the ones that will make or break your projects. Learn what software architects need to achieve—and core disciplines and practices for achieving it Master essential software design principles for addressing function, component separation, and data management See how programming paradigms impose discipline by restricting what developers can do Understand what's critically important and what's merely a "detail" Implement optimal, high-level structures for web, database, thick-client, console, and embedded applications Define appropriate boundaries and layers, and organize components and services See why designs and architectures go wrong, and how to prevent (or fix) these failures Clean Architecture is essential reading for every current or aspiring software architect, systems analyst, system designer, and software manager—and for every programmer who must execute someone else's designs. Register your product for convenient access to downloads, updates, and/or corrections as they become available.

"[The authors] are pioneers. . . . Few in our industry have their breadth of knowledge and experience." —From the Foreword by Dave Thomas, Bedarra Labs Domain-Specific Modeling (DSM) is the latest approach to software development, promising to greatly increase the speed and ease of software creation. Early adopters of DSM have been enjoying productivity increases of 500–1000% in production for over a decade. This book introduces DSM and offers examples from various fields to illustrate to experienced developers how DSM can improve software development in their teams. Two

Bookmark File PDF Implementing Domain Specific Languages With Xtext And Xtend By Bettini Lorenzo 2013 Paperback

authorities in the field explain what DSM is, why it works, and how to successfully create and use a DSM solution to improve productivity and quality. Divided into four parts, the book covers: background and motivation; fundamentals; in-depth examples; and creating DSM solutions. There is an emphasis throughout the book on practical guidelines for implementing DSM, including how to identify the necessary language constructs, how to generate full code from models, and how to provide tool support for a new DSM language. The example cases described in the book are available the book's Website, www.dsmbook.com, along with, an evaluation copy of the MetaEdit+ tool (for Windows, Mac OS X, and Linux), which allows readers to examine and try out the modeling languages and code generators. Domain-Specific Modeling is an essential reference for lead developers, software engineers, architects, methodologists, and technical managers who want to learn how to create a DSM solution and successfully put it into practice.

Implementing Domain-Specific Languages with Xtext and XtendPackt Publishing Ltd

Domain-Specific Languages (DSLs)--languages geared to specific vertical or horizontal areas of interest--are generating growing excitement from software engineers and architects. DSLs bring new agility to the creation and evolution of software, allowing selected design aspects to be expressed in terms much closer to the system requirements than standard program code, significantly reducing

Bookmark File PDF Implementing Domain Specific Languages With Xtext And Xtend By Bettini Lorenzo 2013 Paperback

development costs in large-scale projects and product lines. In this breakthrough book, four leading experts reveal exactly how DSLs work, and how you can make the most of them in your environment. With *Domain-Specific Development with Visual Studio DSL Tools*, you'll begin by mastering DSL concepts and techniques that apply to all platforms. Next, you'll discover how to create and use DSLs with the powerful new Microsoft DSL Tools--a toolset designed by this book's authors. Learn how the DSL Tools integrate into Visual Studio--and how to define DSLs and generate Visual Designers using Visual Studio's built-in modeling technology. In-depth coverage includes Determining whether DSLs will work for you Comparing DSLs with other approaches to model-driven development Defining, tuning, and evolving DSLs: models, presentation, creation, updates, serialization, constraints, validation, and more Creating Visual Designers for new DSLs with little or no coding Multiplying productivity by generating application code from your models with easy-to-use text templates Automatically generating configuration files, resources, and other artifacts Deploying Visual Designers across the organization, quickly and easily Customizing Visual Designers for specialized process needs

List of Figures
List of Tables
Foreword
Preface
About the Authors
Chapter 1 Domain-Specific Development
Chapter 2 Creating and Using DSLs
Chapter 3 Domain Model Definition

Chapter 4 Presentation Chapter 5 Creation, Deletion, and Update Behavior Chapter 6 Serialization Chapter 7 Constraints and Validation Chapter 8 Generating Artifacts Chapter 9 Deploying a DSL Chapter 10 Advanced DSL Customization Chapter 11 Designing a DSL Index

Following the tradition of previous instances of the MoDELS conference series, 11 workshops and two symposia were hosted in 2007. These satellite events complemented the main conference by providing room for important subject areas and enabling a high degree of interactivity. The selection of the workshops was organized like in former instances of the MoDELS conference series by a Workshop Selection Committee. The following well-known experts agreed to serve on this committee: Gabor Karsai, Vanderbilt University, USA Thomas Kuhne, Darmstadt University of Technology, Germany Jochen Kuster, IBM Research Zurich, Switzerland Henry Muccini, University of L'Aquila, Italy Sebastian Uchitel, Imperial College London, UK The workshops provided collaborative forums for particular topics. They enabled a group of participants to exchange recent and/or preliminary results, to conduct intensive discussions, or to coordinate efforts between representatives of a technical community. They served as forums for lively discussion of innovative ideas, recent progress, or practical experience on model-driven engineering for specific

aspects, specific problems, or domain-specific needs. As in previous editions, there were a Doctoral Symposium and an Educators Symposium. The Doctoral Symposium provided specific support for PhD students to discuss their work and receive useful guidance for the completion of their dissertation research. The Educators Symposium addressed how to educate students as well as practitioners to move from traditional thinking to an engineering approach based on models."

The CEFR Companion volume broadens the scope of language education. It reflects academic and societal developments since the publication of the Common European Framework of Reference for Languages (CEFR) and updates the 2001 version. It owes much to the contributions of members of the language teaching profession across Europe and beyond. This volume contains: ? an explanation of the key aspects of the CEFR for teaching and learning; ? a complete set of updated CEFR descriptors that replaces the 2001 set with: - modality-inclusive and gender-neutral descriptors; - added detail on listening and reading; - a new Pre-A1 level, plus enriched description at A1 and C levels; - a replacement scale for phonological competence; - new scales for mediation, online interaction and plurilingual/pluricultural competence; - new scales for sign language competence; ? a short report on the four-year development, validation

and consultation processes. The CEFR Companion volume represents another step in a process of engagement with language education that has been pursued by the Council of Europe since 1971 and which seeks to: ? promote and support the learning and teaching of modern languages; ? enhance intercultural dialogue, and thus mutual understanding, social cohesion and democracy; ? protect linguistic and cultural diversity in Europe; and ? promote the right to quality education for all.

EMF: Eclipse Modeling Framework Dave Steinberg Frank Budinsky Marcelo Paternostro Ed Merks Series Editors: Erich Gamma • Lee Nackman • John Wiegand The Authoritative Guide to EMF Modeling and Code Generation The Eclipse Modeling Framework enables developers to rapidly construct robust applications based on surprisingly simple models. Now, in this thoroughly revised Second Edition, the project's developers offer expert guidance, insight, and examples for solving real-world problems with EMF, accelerating development processes, and improving software quality. This edition contains more than 40% new material, plus updates throughout to make it even more useful and practical. The authors illuminate the key concepts and techniques of EMF modeling, analyze EMF's most important framework classes and generator patterns, guide you through choosing optimal designs, and introduce powerful framework

customizations and programming techniques. Coverage includes • Defining models with Java, UML, XML Schema, and Ecore • NEW: Using extended Ecore modeling to fully unify XML with UML and Java • Generating high-quality code to implement models and editors • Understanding and customizing generated code • Complete documentation of @model Javadoc tags, generator model properties, and resource save and load options • NEW: Leveraging the latest EMF features, including extended metadata, feature maps, EStore, cross-reference adapters, copiers, and content types • NEW: Chapters on change recording, validation, and utilizing EMF in stand-alone and Eclipse RCP applications • NEW: Modeling generics with Ecore and generating Java 5 code About the Authors Dave Steinberg is a software developer in IBM Software Group. He has worked with Eclipse and modeling technologies since joining the company, and has been a committer on the EMF project since its debut in 2002. Frank Budinsky, a senior architect in IBM Software Group, is an original coinventor of EMF and a founding member of the EMF project at Eclipse. He is currently cochair of the Service Data Objects (SDO) specification technical committee at OASIS and lead SDO architect for IBM. Marcelo Paternostro is a software architect and engineer in IBM Software Group. He is an EMF committer and has been an active contributor to several other

Eclipse projects. Before joining IBM, Marcelo managed, designed, and implemented numerous projects using Rational's tools and processes. Ed Merks is the project lead of EMF and a colead of the top-level Modeling project at Eclipse. He holds a Ph.D. in Computing Science and has many years of in-depth experience in the design and implementation of languages, frameworks, and application development environments. Ed works as a software consultant in partnership with itemis AG.

An example-oriented approach to develop custom domain-specific languages. If you've already developed a few Clojure applications and wish to expand your knowledge on Clojure or domain-specific languages in general, then this book is for you. If you're an absolute Clojure beginner, then you may only find the detailed examples of the core Clojure components of value. If you've developed DSLs in other languages, this Lisp and Java-based book might surprise you with the power of Clojure.

What others in the trenches say about The Pragmatic Programmer... "The cool thing about this book is that it's great for keeping the programming process fresh. The book helps you to continue to grow and clearly comes from people who have been there." —Kent Beck, author of *Extreme Programming Explained: Embrace Change* "I found this book to be a great mix of solid advice and wonderful analogies!" —Martin Fowler, author of *Refactoring and UML*

Bookmark File PDF Implementing Domain Specific Languages With Xtext And Xtend By Bettini Lorenzo 2013 Paperback

Distilled “I would buy a copy, read it twice, then tell all my colleagues to run out and grab a copy. This is a book I would never loan because I would worry about it being lost.” —Kevin Ruland, Management Science, MSG-Logistics “The wisdom and practical experience of the authors is obvious. The topics presented are relevant and useful.... By far its greatest strength for me has been the outstanding analogies—tracer bullets, broken windows, and the fabulous helicopter-based explanation of the need for orthogonality, especially in a crisis situation. I have little doubt that this book will eventually become an excellent source of useful information for journeymen programmers and expert mentors alike.” —John Lakos, author of Large-Scale C++ Software Design “This is the sort of book I will buy a dozen copies of when it comes out so I can give it to my clients.” —Eric Vought, Software Engineer “Most modern books on software development fail to cover the basics of what makes a great software developer, instead spending their time on syntax or technology where in reality the greatest leverage possible for any software team is in having talented developers who really know their craft well. An excellent book.” —Pete McBreen, Independent Consultant “Since reading this book, I have implemented many of the practical suggestions and tips it contains. Across the board, they have saved my company time and money while helping me get

my job done quicker! This should be a desktop reference for everyone who works with code for a living.” —Jared Richardson, Senior Software Developer, iRenaissance, Inc. “I would like to see this issued to every new employee at my company....” —Chris Cleeland, Senior Software Engineer, Object Computing, Inc. “If I’m putting together a project, it’s the authors of this book that I want. . . . And failing that I’d settle for people who’ve read their book.” —Ward Cunningham

Straight from the programming trenches, *The Pragmatic Programmer* cuts through the increasing specialization and technicalities of modern software development to examine the core process--taking a requirement and producing working, maintainable code that delights its users. It covers topics ranging from personal responsibility and career development to architectural techniques for keeping your code flexible and easy to adapt and reuse. Read this book, and you'll learn how to Fight software rot; Avoid the trap of duplicating knowledge; Write flexible, dynamic, and adaptable code; Avoid programming by coincidence; Bullet-proof your code with contracts, assertions, and exceptions; Capture real requirements; Test ruthlessly and effectively; Delight your users; Build teams of pragmatic programmers; and Make your developments more precise with automation. Written as a series of self-contained sections and filled with entertaining

anecdotes, thoughtful examples, and interesting analogies, *The Pragmatic Programmer* illustrates the best practices and major pitfalls of many different aspects of software development. Whether you're a new coder, an experienced programmer, or a manager responsible for software projects, use these lessons daily, and you'll quickly see improvements in personal productivity, accuracy, and job satisfaction. You'll learn skills and develop habits and attitudes that form the foundation for long-term success in your career. You'll become a Pragmatic Programmer.

A general-purpose language like C# is designed to handle all programming tasks. By contrast, the structure and syntax of a Domain-Specific Language are designed to match a particular applications area. A DSL is designed for readability and easy programming of repeating problems. Using the innovative Boo language, it's a breeze to create a DSL for your application domain that works on .NET and does not sacrifice performance. *DSLs in Boo* shows you how to design, extend, and evolve DSLs for .NET by focusing on approaches and patterns. You learn to define an app in terms that match the domain, and to use Boo to build DSLs that generate efficient executables. And you won't deal with the awkward XML-laden syntax many DSLs require. The book concentrates on writing internal (textual) DSLs that allow easy extensibility of the application and

Bookmark File PDF Implementing Domain
Specific Languages With Xtext And Xtend By
Bettini Lorenzo 2013 Paperback

framework. And if you don't know Boo, don't worry-
you'll learn right here all the techniques you need.
Purchase of the print book comes with an offer of a
free PDF, ePub, and Kindle eBook from Manning.
Also available is all code from the book.

[Copyright: 3aeceeff178105afdecccb027d74923](#)