

1. Record Nr.	UNINA9910816475903321
Titolo	Patterns : model-driven development using IBM Rational Software Architect / / Peter Swithinbank ... [et al.]
Pubbl/distr/stampa	Research Triangle Park, N.C., : IBM, International Technical Support Organization, c2005
Edizione	[1st ed.]
Descrizione fisica	xvi, 236 p. : ill
Collana	Redbooks
Disciplina	005.1
Soggetti	Computer software - Development Software architecture UML (Computer science)
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	"This edition applies to Version 6.0.0.1 of Rational Software Architect (product number 5724-I70)." "December 2005."
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Front cover -- Contents -- Notices -- Trademarks -- Preface -- For solution architects -- For project planners or project managers -- For those working on a project that uses model-driven development -- How this book is organized -- The team that wrote this redbook -- Become a published author -- Comments welcome -- Part 1 Approach -- Chapter 1. Overview and concepts of model-driven development -- 1.1 Current business environment and drivers -- 1.2 A model-driven approach to software development -- 1.2.1 Models as sketches and blueprints -- 1.2.2 Precise models enable automation -- 1.2.3 The role of patterns in model-driven development -- 1.2.4 Not just code -- 1.3 Benefits of model-driven development -- 1.4 Model-driven development with IBM Rational Software Architect -- 1.4.1 Unified Modeling Language 2.0 editor -- 1.4.2 UML profile support -- 1.4.3 RSA patterns -- 1.4.4 RSA transformations -- 1.5 Summary -- Chapter 2. Scenario overview -- 2.1 Enterprise architecture -- 2.1.1 Suitability for model-driven development -- 2.1.2 Contra-indications for model-driven development -- 2.2 Integration architecture -- 2.2.1 ESB structure -- 2.3 Pattern definition -- 2.3.1 Interaction behavior patterns -- 2.3.2 Individual service patterns -- 2.3.3 Suitability for

model-driven development -- 2.3.4 Contra-indications for model-driven development -- 2.4 Automation -- 2.4.1 Technical -- 2.4.2 Organizational -- 2.4.3 Managerial -- 2.5 Summary -- Chapter 3. Model-driven development approach -- 3.1 Abstraction -- 3.2 Precise modeling -- 3.3 Automation -- 3.4 Architectural style -- 3.5 The role of UML -- 3.6 Expertise capture -- 3.6.1 Logical architecture expertise -- 3.6.2 Technical architecture expertise -- 3.7 Patterns -- 3.8 Quality and consistency -- 3.9 Integration -- 3.10 Platform independence -- 3.11 Layered modeling. 3.12 Modeling of non-functional characteristics -- 3.13 Summary -- Chapter 4. Model-driven development project planning -- 4.1 The value and cost of model-driven development -- 4.2 Understanding the tasks for a model-driven development project -- 4.2.1 Descriptions of tasks -- 4.2.2 The model-driven development tool chain -- 4.3 Planning a model-driven development project -- 4.3.1 Using an iterative approach to model-driven development -- 4.3.2 Developing model-driven development skills -- 4.3.3 Thinking about reuse -- 4.4 Quality control for model-driven development tooling -- 4.5 Tracking a model-driven development project -- 4.6 At the end of the project -- 4.7 Summary -- Chapter 5. Model-driven development solution life cycle -- 5.1 Introduction to the solution life cycle -- 5.2 Model-driven development life cycle -- 5.2.1 Create the framework to generate the solution services -- 5.2.2 Generate, customize, and test the solution services -- 5.3 Model-driven development and versioning -- 5.3.1 Versioning and replacement policies -- 5.4 Model-driven development and artifact management -- 5.4.1 Reuse model artifacts -- 5.4.2 Integrity management services -- 5.4.3 Deployment support -- 5.5 Model-driven development and problem determination -- 5.5.1 Tooling versus instrumentation -- 5.6 Information mining -- 5.7 Testing -- 5.7.1 Modeling for testing -- 5.7.2 Applying test patterns -- 5.7.3 Modeling using the UML testing profile -- 5.8 Summary -- Chapter 6. Model-driven development in context -- 6.1 OMG and Model-Driven Architecture -- 6.2 MDA models -- 6.2.1 IBM and MDA -- 6.3 Software Factories and domain-specific languages -- 6.3.1 UML and DSLs -- 6.4 Asset-based development -- 6.5 Pattern-driven development and IBM Patterns for e-business -- 6.5.1 IBM Patterns for e-business -- 6.6 Business-driven development. 6.7 Model-driven development and On Demand Business -- 6.8 Model-driven development and middleware -- 6.9 Visualization -- 6.10 Executable UML -- 6.11 Summary -- Part 2 Implementation -- Chapter 7. Designing patterns for the scenario -- 7.1 Relationship to the project plan -- 7.2 Overview of pattern design -- 7.3 Architecture patterns -- 7.4 Contracts of behavior -- 7.4.1 Contract of behavior for synchronous updates -- 7.4.2 General requirements for synchronous update -- 7.5 Integration patterns -- 7.6 Applying a pattern to create a high-level model -- 7.6.1 The pattern -- 7.6.2 The model -- 7.7 Detailing the initial model with service patterns -- 7.7.1 Service patterns: Activity diagrams -- 7.7.2 Integration services -- 7.8 RSA transformation -- 7.8.1 Implementing the integration facade -- 7.8.2 Implementing the integration service -- 7.8.3 Implementing the provider facade -- 7.9 Use of the framework -- 7.9.1 Presentation of model information to users -- 7.9.2 Service creation -- 7.10 Summary -- Chapter 8. Applying model-driven development with Rational Software Architect -- 8.1 An overview of the Model-driven development process in RSA -- 8.1.1 Framework development -- 8.1.2 Application development -- 8.2 RSA model-driven development framework for SOI -- 8.3 Application development -- 8.3.1 Installing the framework -- 8.3.2 Creating a model and apply the profiles --

8.3.3 Applying patterns -- 8.3.4 Applying transformations -- 8.3.5  
Testing the generated code -- 8.3.6 Application development summary  
-- 8.4 Framework development -- 8.4.1 Developing the architectural  
style -- 8.4.2 Creating a UML profile -- 8.4.3 Implementing sample  
components -- 8.4.4 Developing patterns and transformations -- 8.5  
Summary -- Chapter 9. Extending Rational Software Architect -- 9.1  
Introduction to implementing patterns and transformations to RSA.  
9.2 Setup: Enabling Eclipse Developer -- 9.3 Deploying UML profiles --  
9.3.1 Defining a path map -- 9.3.2 Releasing the profile -- 9.3.3  
Adding the profile to a plug-in -- 9.3.4 Deploying the plug-in -- 9.4  
Implementing patterns -- 9.4.1 Getting started -- 9.4.2 Defining a  
pattern -- 9.4.3 Pattern implementation -- 9.4.4 Testing the pattern --  
9.4.5 Publishing patterns -- 9.5 Implementing a transformation --  
9.5.1 Creating a new plug-in with a transformation -- 9.5.2  
Transformation API -- 9.5.3 Implementing the root transformation --  
9.5.4 Implementing the transformation rules -- 9.5.5 Creating and  
modifying files in the RSA workspace -- 9.5.6 Testing the  
transformation -- 9.6 Launching a Run-time Workbench -- 9.7  
Deploying plug-ins -- 9.8 Using a RAS repository -- 9.9 Summary --  
Chapter 10. Conclusion -- Appendix A. Additional material -- Locating  
the Web material -- Using the Web material -- System requirements for  
downloading the Web material -- How to use the Web material --  
Abbreviations and acronyms -- Related publications -- IBM Redbooks  
-- Other publications -- Online resources -- How to get IBM Redbooks  
-- Help from IBM -- Index -- Back cover.

---