

ANNEKE KLEPPE
JOS WARMER
WIM BAST

MDA EXPLAINED

The Model Driven Architecture™: Practice and Promise

“Jos Warmer’s work has contributed greatly to the semantics of the UML. From that perspective, and in this book, he offers insight on how one can and can’t use the UML to move to the next level of abstraction in building systems.”

—Grady Booch

Experienced application developers often invest more time in building models than they do in actually writing code. Why? Well-constructed models make it easier to deliver large, complex enterprise systems on time and within budget. Now, a new framework advanced by the Object Management Group (OMG) allows developers to build systems according to their core business logic and data—independently of any particular hardware, operating system, or middleware.

Model Driven Architecture (MDA) is a framework based on the Unified Modeling Language (UML) and other industry standards for visualizing, storing, and exchanging software designs and models. However, unlike UML, MDA promotes the creation of machine-readable, highly abstract models that are developed independently of the implementation technology and stored in standardized repositories. There, they can be accessed repeatedly and automatically transformed by tools into schemas, code skeletons, test harnesses, integration code, and deployment scripts for various platforms.

Written by three members of OMG’s MDA standardization committee, *MDA Explained* gives readers an inside look at the advantages of MDA and how they can be realized.

This book begins with practical examples that illustrate the application of different types of models. It then shifts to a discussion at the meta-level, where developers will gain the knowledge necessary to define MDA tools.

Highlights of this book include:

- The MDA framework, including the Platform Independent Model (PIM) and Platform Specific Model (PSM)
- OMG standards and the use of UML
- MDA and Agile, Extreme Programming, and Rational Unified Process (RUP) development
- How to apply MDA, including PIM-to-PSM and PSM-to-code transformations for Relational, Enterprise JavaBean (EJB), and Web models
- Transformations, including controlling and tuning, traceability, incremental consistency, and their implications
- Metamodeling
- Relationships between different standards, including Meta Object Facility (MOF), UML, and Object Constraint Language (OCL)

The advent of MDA offers concrete ways to improve productivity, portability, interoperability, maintenance, and documentation dramatically. With this groundbreaking book, IT professionals can learn to tap this new framework to deliver enterprise systems most efficiently.

©2003, PAPERBACK, 192 PAGES,
0-321-19442-X, \$34.99

ABOUT THE AUTHORS

ANNEKE KLEPPE, one of the principal developers of the Object Constraint Language, founded Klasse Objecten in 1995.

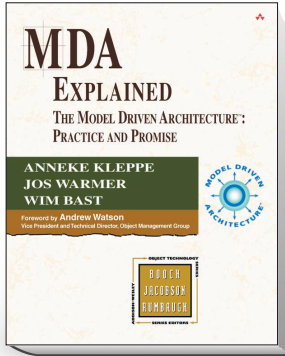
JOS WARMER is a consultant and adviser with Klasse Objecten. As an active member of the Unified Modeling Language Revision Task Force, he is working on revisions that will become part of the UML 2.0 standard.

WIM BAST is Compuware’s OptimalJ MDA architect.


Addison
Wesley

FROM ADDISON-WESLEY

TABLE OF CONTENTS



1: The MDA Development Process

Traditional Software Development
 • The Productivity Problem • The Portability Problem • The Interoperability Problem • The Maintenance and Documentation Problem • The Model Driven Architecture • The MDA Development Life Cycle • Automation of the Transformation Steps • MDA Benefits • Productivity • Portability • Interoperability • Maintenance and Documentation • MDA Building Blocks • Summary

2: The MDA Framework

What Is a Model? • Relationships between Models • Types of Models • Business and Software Models • Structural and Dynamic Model • Platform Independent and Platform Specific Models • The Target Platforms of a Model • What is a Transformation? • Transformations between Identical Languages • The Basic MDA Framework • Examples • Public and Private Attributes • Associations • Summary

Chapter 3: MDA Today

OMG Standards • OMG Languages • OMG Language and Transformation Definitions • UML as PIM Language • Plain UML • Executable UML • UML<ETH>OCL Combination • Tools • Support for Transformations • Categorizing Tools • Development Processes • Agile Software Development • Extreme Programming • Rational Unified Process (RUP) • Summary

4: Rosa's Application of MDA

Rosa's Breakfast Service • The Business • The Software System • Applying the MDA Framework • The PIM and PSMs • The PIM to PSM Transformations • The PSM to Code Model Transformations • Three Levels of Abstraction • The PIM in Detail • Summary

5: Rosa's PIM to Three PSMs

The PIM to Relational Transformation • The PIM to EJB Transformation • A Coarse Grained EJB Model • The Transformation Rules • The PIM to Web Transformation • The Transformation Rules • The Communication Bridges • Summary

6: Rosa's PSMs to Code

Relational Model to Code Transformation • EJB Model to Code Transformation • Some Remarks on EJB Code • The Transformation Rules • The Web Model to Code Transformation • The Structure of the Web Code • The Transformation Rules • Summary

7: More on Transformations

Desired Features of Transformations • Controlling and Tuning Transformations • Manual Control • Conditions on Transformations • Transformation Parameters • Additional Information • Traceability • Incremental Consistency • Bidirectionality • Implications on Transformations • Transformation Parameters • Persistent Source-Target Relationship • Transformation Rules as Objects • Summary

8: Metamodeling

Introduction to Metamodeling • The Four Modeling Layers of the OMG • Layer M0: The Instances • Layer M1: The Model of the System • Layer M2: The Model of the Model • Layer M3: The Model of M2 • Getting Rid of the Layers • The Use of Metamodeling in the MDA • The Extended MDA Framework • Summary

9: Defining Your Own Transformations

Transformations Definitions Revisited • The Transformation Definition Language • Requirements for a Transformation Rule • A Notation for Transformation Rules • Transformation Definitions • Example Transformation Definitions • Public and Private Attributes • Associations • Classes • Finishing the Transformation Definition • The Complete MDA Framework • Summary

10: Rosa's Transformation Definitions

The UML to Relational Mapping • Transformation Rules for UML to Relational Model • Completion of the Relational Model • The UML to EJB Mapping • Additional Mappings • The UML to Web Mapping • Summary

11: OMG Standards and Additional Technologies

Introduction • The MOF • yMOF Tools • The Role of the MOF in MDA • Query, Views, and Transformations • UML • The UML Metamodel • The Role of UML in MDA • OCL • Using OCL with UML • Using OCL with the MOF • The Role of OCL in MDA • The UML Action Semantics • CWM • UML Profiles • The Role of Profiles in MDA • Summary

12: The MDA Promise

The MDA Paradigm Shift • A Historic Perspective • A Shift of Focus • Too Good to Be True? • The Development Process • The Tools • The Modeling Languages • Summary

A: Glossary

B: The Code for Rosa's System

The SQL Code for Rosa's System • The EJB Code for Rosa's System • The JSP Code for Rosa's System

ORDERING INFORMATION:

SINGLE COPY SALES:

Visa, Master Card, American Express, Checks, or Money Orders only —
 Tel: 515-284-6761
 Fax: 515-237-4416
 Toll-Free:
 800-282-0693

GOVERNMENT AGENCIES:

Kathryn Bass
 GS-14F-8023A
 703-404-9194
 www.pearsongovernmentales.com

COLLEGE PROFESSORS:

Desk or Review Copies —
 exam@aw.com

CORPORATE ACCOUNTS:

Quantity, Bulk Orders totalling 10 or more books. Purchase orders only —
 No credit cards.
 Fax: 317-817-7307
 Toll-Free:
 800-382-3419

INTERNATIONAL ORDERING INFORMATION:

CANADA:
 beverley.carkner@pearsoned.com

UK/EMEA:
 Europe, Middle East, South Africa
 jonathan.hardy@pearsoned-ema.com

BENELUX:
 raymond.ashruf@pearsoned-ema.com

AUSTRALIA:
 trade@pearsoned.com.au

SOUTH ASIA:
 leehong.tan@pearsoned.com.sg

NORTH ASIA:
 cheranL@pearsoned.com.hk

OTHER REGIONS:
 sarah.hlibka@pearsoned.com