Common Warehouse Metamodel Developer’s Guide

John Poole, Dan Chang, Douglas Tolbert, and David Mellor

Wiley Publishing, Inc.
Dear Valued Customer,

We realize you’re a busy professional with deadlines to hit. Whether your goal is to learn a new technology or solve a critical problem, we want to be there to lend you a hand. Our primary objective is to provide you with the insight and knowledge you need to stay atop the highly competitive and ever-changing technology industry.

Wiley Publishing, Inc., offers books on a wide variety of technical categories, including security, data warehousing, software development tools, and networking — everything you need to reach your peak. Regardless of your level of expertise, the Wiley family of books has you covered.

- For Dummies® – The fun and easy way® to learn
- The Weekend Crash Course® – The fastest way to learn a new tool or technology
- Visual™ – For those who prefer to learn a new topic visually
- The Bible – The 100% comprehensive tutorial and reference
- The Wiley Professional list – Practical and reliable resources for IT professionals

The book you now hold, Common Warehouse Metamodel Developer’s Guide, is your complete and authoritative guide to developing datawarehousing and business intelligence applications via the Common Warehouse Metamodel (CWM) framework. Written by several of the core developers of the CWM standard, the book will show you all the steps you’ll need for planning and implementing a CWM-enabled datawarehousing environment. The authors provide you with detailed guidelines and in-depth code examples that will allow you to put the Common Warehouse Metamodel to work in your business.

Our commitment to you does not end at the last page of this book. We’d want to open a dialog with you to see what other solutions we can provide. Please be sure to visit us at www.wiley.com/compbooks to review our complete title list and explore the other resources we offer. If you have a comment, suggestion, or any other inquiry, please locate the “contact us” link at www.wiley.com.

Finally, we encourage you to review the following page for a list of Wiley titles on related topics. Thank you for your support and we look forward to hearing from you and serving your needs again in the future.

Sincerely,

Richard K. Swadley
Vice President & Executive Group Publisher
Wiley Technology Publishing

Wiley, For Dummies, Weekend Crash Course, Visual and related trademarks, logos and trade dress are trademarks of Wiley Publishing, Inc. All other trademarks are property of their respective owners.

more information on related titles
Also from OMG Press

Available from Wiley Publishing

Mastering XMI
Java Programming with XMI, XML, and UML
Timothy J. Grove, Gary C. Dunnet, Stephen A. Brodky, PhD

Enterprise Java with UML
CT Arrington

Common Warehouse Metamodel
An Introduction to the Standard for Data Warehouse Integration
John Poole, Dan Chang, Douglas Tilbert, and David Melter

Model Driven Architecture
Applying MDA to Enterprise Computing
David S. Frankel

Common Warehouse Metamodel
Developer’s Guide
John Poole, Dan Chang, Douglas Tilbert, David Melter

Available at your favorite bookseller or visit www.wiley.com/compbooks
Common Warehouse Metamodel Developer’s Guide

John Poole, Dan Chang, Douglas Tolbert, and David Mellor
Advance Praise for
Common Warehouse
Metamodel Developer’s Guide

“CWM Developer’s Guide breaks new ground by providing an in-depth overview of Model Driven Integration for the data warehouse and Business Intelligence tool chain using innovative meta data design patterns. The use of UML and MOF to define platform-independent models while simultaneously targeting both XML and Java-based meta data management using XMI and JMI is supported with numerous examples. Software architects, CTOs, systems integrators, and vendors grappling with the complexity of tool, data, and application integration can learn firsthand the power of OMG Model Driven Architecture from this pioneering book.”

Sridhar Iyengar
IBM Distinguished Engineer, OMG Architecture Board

“The first CWM book, Common Warehouse Metamodel: An Introduction, has become a great complement to the CWM specifications. This follow-on book delves even deeper into the implementation world, which is critical to the success of any standard. This developer’s guide establishes a key transition from ‘paper standards’ to actual adopted standards for tool integration.

Common Warehouse Metamodel Developer’s Guide is not only well written, but also well focused on applications related to the standards. I highly recommend this second book to anyone who wants to transform standards into reality in their product strategy.”

Christian H. Bremeau
President and CEO, Meta Integration Technology, Inc. (MITI)
Common Warehouse Metamodel Developer’s Guide is a highly practical guide to a powerful new way of integrating systems in the data warehousing and business analysis domains. By leveraging this new standard for modeling and exchanging application, tool, and instance meta data, the authors show how representing common business and domain concepts as higher-level abstractions can solve complex, real-world integration problems.

Model-based development has the potential to vastly simplify the increasingly complex issues faced by developers in building integrated solutions in today’s distributed, heterogeneous environments, and CWM is the leading example of the success of this approach.”

Chuck Mosher
Staff Engineer, Market Development Engineering, Sun Microsystems

“This book illustrates how CWM is used not only to describe complex data warehousing systems, but also to facilitate interoperability and integration. It is an excellent guide for anyone interested in developing platform-independent domain models and leveraging domain models for integration and information exchange.”

Ravi Dirckze
JMI 1.0 Specification Lead and Senior Software Engineer, Unisys Corporation
OMG Advisory Board

David Frankel  
Chief Consulting Architect  
IONA

Sridhar Iyengar  
Distinguished Engineer  
IBM Corporation

Cris Kobryn  
Chief Technologist  
Telelogic

Nilo Mitra, Ph.D.  
Principal System Engineer  
Ericsson

Jishnu Mukerji  
Senior Systems Architect  
Hewlett-Packard Company

Jon Siegel, Ph.D.  
Vice President, Technology Transfer  
Object Management Group, Inc.

Richard Mark Soley, Ph.D.  
Chairman and Chief Executive Officer  
Object Management Group, Inc.
The Object Management Group (OMG) is an open membership, not-for-profit consortium that produces and maintains computer industry specifications for interoperable applications. To achieve this goal, the OMG specifies open standards for every aspect of distributed computing from analysis and design, through infrastructure, to application objects and components defined on virtually every enterprise middleware platform. OMG’s membership roster includes virtually every large company in the computer industry, and hundreds of smaller ones. Most of the companies that shape enterprise and Internet computing today are represented on OMG’s Board of Directors.

OMG’s flagship specification, and the basis for future OMG specifications, is the multi-platform Model Driven Architecture (MDA). Unifying the modeling and middleware spaces, the MDA supports applications over their entire life-cycle from Analysis and Design, through implementation and deployment, to maintenance and evolution. Based on normative, platform-independent Unified Modeling Language (UML) models, MDA-based applications and standards may be expressed and implemented, equivalently, on multiple middleware platforms; implementations are produced automatically, for the most part, by MDA-enabled tools, which also generate cross-platform invocations making for a truly interoperable environment. Because the UML models remain stable as the technological landscape changes around them over time, MDA-based development maximizes software ROI as it integrates applications across the enterprise, and one enterprise with another. Adopted by members as the basis for OMG specifications in September 2001, the MDA is truly a unique advance in distributed computing. To learn more about the MDA, see www.omg.org/mda.

OMG’s modeling specifications form the foundation for the MDA. These include the UML, the MetaObject Facility (MOF), XML Metadata Interchange
(XMI), and the Common Warehouse Metamodel (CWM). The industry’s standard for representation of analysis and design, the UML defines Use Case and Activity diagrams for requirements gathering, Class and Object diagrams for design, Package and Subsystem diagrams for deployment, and six other diagram types. The MOF defines a standard metamodel for applications, allowing UML models to be interchanged among tools and repositories; and XMI standardizes the format for these interchanges. Finally, CWM establishes metamodels in the field of data warehousing, completing OMG’s standardization in the modeling space.

The Common Object Request Broker Architecture (CORBA) is OMG’s vendor-neutral, system-independent middleware standard. Based on the OMG/ISO Interface Definition language (OMG IDL) and the Internet InterORB Protocol (IIOP), CORBA is a mature technology represented on the market by more than 70 ORBs (Object Request Brokers) plus hundreds of other products. Scalable to Internet and Enterprise levels, CORBA more than meets business computing requirements through its robust services providing directory, distributed event handling, transactionality, fault tolerance, and security. Specialized versions of CORBA form the basis for distributed Realtime computing, and distributed embedded systems.

Building on this foundation, OMG Domain Facilities standardize common objects throughout the supply and service chains in industries such as Telecommunications, Healthcare, Manufacturing, Transportation, Finance/Insurance, Biotechnology, Utilities, Space, and Military and Civil Defense Logistics. OMG members are now extending these Domain Facilities, originally written in OMG IDL and restricted to CORBA, into the MDA by constructing UML models corresponding to their underlying architecture; standard MDA procedures will then produce standards and implementations on such platforms as Web Services, XML/SOAP, Enterprise JavaBeans, and others. OMG’s first MDA-based specification, the Gene Expression Facility, was adopted less than six months after the organization embraced the MDA; based on a detailed UML model, this specification is implemented entirely in the popular language XML.

In summary, the OMG provides the computing industry with an open, vendor-neutral, proven process for establishing and promoting standards. OMG makes all of its specifications available without charge from its Web site, www.omg.org. Delegates from the hundreds of OMG member companies convene at week-long meetings held five times each year at varying sites around the world, to advance OMG technologies. The OMG welcomes guests to their meetings; for an invitation, send your email request to info@omg.org or see www.omg.org/news/meetings/tc/guest.htm.

Membership in OMG is open to any company, educational institution, or government agency. For more information on the OMG, contact OMG headquarters by telephone at +1-781-444-0404, by fax at +1-781-444-0320, by email to info@omg.org, or on the Web at www.omg.org.
John: To Robert J. Flynn, teacher and friend.

Dan: To Don Haderle and Jo Chang, without whose vision and support CWM would not exist.

Doug: For my parents, Ken & Jeanetta, who started all this for me.

David: To my wife Michelle, my daughter Marie, my mother Annette, and my sister and brother, Debbie and Roger.
Acknowledgments xxiii
Introduction xxv
About the Authors xxxiii

Part 1 Introduction

Chapter 1 Introducing CWM: Model-Based Integration of the Supply Chain 3
  Integrating the Information Supply Chain 4
  Components of the Information Supply Chain 4
  The Economics of Integrating the ISC 6
  CWM: Model-Based Meta Data Integration 10
  The Model-Based Approach to Meta Data 10
  An Overview of CWM 16
  Summary 22

Chapter 2 An Architectural Survey of CWM 25
  The CWM Metamodel Packages 26
    The Object Model Layer 27
      The Core Package 27
      Behavioral Package 31
      Relationships Package 31
      Instance Package 33
    Foundation Layer 34
      Business Information Package 34
      DataTypes Package 36
      Expressions Package 39
      Keys and Indexes Package 41
### Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Deployment Package</td>
<td>43</td>
</tr>
<tr>
<td>TypeMapping Package</td>
<td>44</td>
</tr>
<tr>
<td>Resource Layer</td>
<td>46</td>
</tr>
<tr>
<td>Object Package</td>
<td>46</td>
</tr>
<tr>
<td>Relational Package</td>
<td>47</td>
</tr>
<tr>
<td>Record Package</td>
<td>49</td>
</tr>
<tr>
<td>Multidimensional Package</td>
<td>50</td>
</tr>
<tr>
<td>XML Package</td>
<td>51</td>
</tr>
<tr>
<td>Analysis Layer</td>
<td>53</td>
</tr>
<tr>
<td>Transformation Package</td>
<td>54</td>
</tr>
<tr>
<td>OLAP Package</td>
<td>57</td>
</tr>
<tr>
<td>Data Mining Package</td>
<td>59</td>
</tr>
<tr>
<td>Information Visualization Package</td>
<td>60</td>
</tr>
<tr>
<td>Business Nomenclature Package</td>
<td>62</td>
</tr>
<tr>
<td>Management Layer</td>
<td>64</td>
</tr>
<tr>
<td>Warehouse Process Package</td>
<td>64</td>
</tr>
<tr>
<td>Warehouse Operation Package</td>
<td>64</td>
</tr>
<tr>
<td>Key Architectural Concepts: Extending CWM</td>
<td>66</td>
</tr>
<tr>
<td>Meta Data Reuse and Extension Based on Inheritance</td>
<td>67</td>
</tr>
<tr>
<td>Lightweight Extension Mechanisms: Stereotypes and TaggedValues</td>
<td>71</td>
</tr>
<tr>
<td>Summary</td>
<td>73</td>
</tr>
</tbody>
</table>

#### Chapter 3  Modeling Meta Data Using CWM

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>UML</td>
<td>77</td>
</tr>
<tr>
<td>Building Blocks and Well-Formedness Rules</td>
<td>77</td>
</tr>
<tr>
<td>Static Structure Modeling</td>
<td>80</td>
</tr>
<tr>
<td>Model Management</td>
<td>82</td>
</tr>
<tr>
<td>Meta Object Facility</td>
<td>83</td>
</tr>
<tr>
<td>The MOF Model</td>
<td>84</td>
</tr>
<tr>
<td>The CWM Metamodel</td>
<td>87</td>
</tr>
<tr>
<td>How CWM Uses Inheritance to Achieve Reuse</td>
<td>89</td>
</tr>
<tr>
<td>How Meta Data Links to Physical Data Resources</td>
<td>91</td>
</tr>
<tr>
<td>How Resource Packages Support Instance Objects</td>
<td>93</td>
</tr>
<tr>
<td>Using CWM to Model Meta Data</td>
<td>94</td>
</tr>
<tr>
<td>Modeling Relational Meta Data</td>
<td>97</td>
</tr>
<tr>
<td>Modeling Record-Based Meta Data</td>
<td>101</td>
</tr>
<tr>
<td>Modeling Physical Data Resources</td>
<td>106</td>
</tr>
<tr>
<td>Modeling Transformation Meta Data</td>
<td>108</td>
</tr>
<tr>
<td>Modeling OLAP Meta Data</td>
<td>112</td>
</tr>
<tr>
<td>Summary</td>
<td>124</td>
</tr>
</tbody>
</table>

#### Chapter 4  Meta Data Interchange Patterns

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introducing Meta Data Interchange Patterns</td>
<td>126</td>
</tr>
<tr>
<td>The Need to Establish a Common Context for Meta Data Interchange</td>
<td>127</td>
</tr>
<tr>
<td>The Need to Place Boundaries on Solution Extents</td>
<td>134</td>
</tr>
</tbody>
</table>
Chapter 6  **Dimensional Model**  235

The Logical Model  
Dimensions, Attributes, Levels, and Hierarchies  237
The CWM Model  238
Defining the Dimensions and Attributes  240
Defining Levels and Level Attributes  244
Defining Hierarchies and Hierarchical Attributes  251
Add the Dimensions to the Schema  263
Defining Cubes and Measures  264
Add the Cubes to the Schema  267
Defining Keys  268
Adding Keys to the Dimensions  269
Adding Keys to the Levels  270
Adding Keys to the Hierarchies  271
Adding Keys to the Cubes  277
The Physical Model  279
A Relational Star-Schema  280
Defining the Physical Objects  280
Defining the Tables and Columns  280
Adding Primary Keys and Foreign Keys  288
Physical Deployment Models  294
Creating the DeploymentGroup  295
The CWM Mapping Model  296
Mapping the Logical Model  297
Mapping the Physical Model  312
Creating a Second Deployment 336
The Multidimensional Metamodel 336
The Express Model 336
Creating the Express Objects 338
Adding the Second Deployment 339
Summary 347

Chapter 7  **Web-Enabled Data Warehouse Model**  349
Introducing the Web-Enabled Data Warehouse 350
Merging the Web and the Data Warehouse 351
Web-Enabled Dimensional Model 354
The Logical Clickstream Dimensions 354
CWM Packages and Interfaces 356
Building the Time Dimension 357
Building the Customer Dimension 364
Building the Product Dimension 369
Building the Page Dimension 373
Building the Event Dimension 377
Building the Session Dimension 381
Building the Referral Dimension 385
Building the Causal Dimension 390
Building the Entity Dimension 394
The Logical Clickstream Analysis Cubes 400
Session Analysis Cube 401
Page Analysis Cube 405
Aggregation Cube 407
New Meta Data Patterns Developed and Cataloged 411
Local Stereotype, Version 1.0 411
URL 411
Contributor 411
Structural Classification 411
Usage Category 411
Intent 412
Also Known As 412
Motivation 412
Applicability 412
Projection 412
Restriction 413
Usage 413
Parameters 414
Commentary 414
Consequences 414
Known Uses 414
Related Patterns 414
Sample Solution 414
Chapter 8  CWM Metastore

Building a CWM Metastore 434
Object-to-Relational Mapping Patterns 436
  A Field Guide to Object-to-Relational Mapping Patterns 438
    UML Notation Overview 438
    Relational Table Notation 439
    Manipulating Data in Metastore Tables 442
    Preserving CWM’s Object Structure 444
Data Type Mapping Patterns 447
  Simple Data Type Pattern 448
  Enumerated Data Type Pattern 448
  Class-based Data Type Pattern 452
Class Mapping Patterns 453
  Attribute Mapping Patterns 456
  Inheritance Pattern 468
Association Mapping Patterns 487
  One-to-One Association Pattern 491
  One-to-Many Association Pattern 495
  Many-to-Many Association Pattern 500
Using Association Patterns 504
  Enforcing Association Multiplicities 531
Reference Mapping Patterns 532
MetaStore Services 538
  Transaction Management Services 539
  ClassMap Service 540
  AllOfType Services 541
  Error-Handling Services 543
Using the Metastore in Applications 544
Summary 554

Part 3  Implementation and Deployment

Chapter 9  Integration Architecture

Developing a Meta Data Integration Architecture 558
  Survey of Architectural Patterns 558
    Meta Data Interconnection Architecture 559
    Meta Data Life-Cycle Architecture 567
## Contents

CWM-Based Meta Data Integration Architecture 577  
Crafting Your Own CWM Architectural Solutions 583  
Summary 583  

### Chapter 10 Interface Rendering 585  
CWM Core Classes and JMI Mappings 586  
ModelElement, Namespace, and Package 586  
Classifier, Class, Datatype, and Attribute 589  
Method and Parameter 592  
Instances 595  
Keys and Indexes 598  
CWM Relational Classes and JMI Mappings 602  
Catalog and Schema 602  
Table, View, QueryColumnSet, and Column 604  
UniqueConstraint, PrimaryKey, and ForeignKey 610  
SQLIndex and SQLIndexColumn 612  
SQL Data Types 614  
Stored Procedure 618  
Trigger 620  
Relational Instances 622  
Relational Package Proxy 623  
CWM Transformation Classes and JMI Mappings 624  
Transformation 625  
TransformationTask, TransformationStep, and TransformationActivity 628  
TransformationMap and Its Components 633  
CWM WarehouseProcess Classes and JMI Mappings 638  
Summary 642  

### Chapter 11 Implementation Development 643  
CWM Implementation 643  
Extending CWM 645  
Simple Extensions to CWM 645  
Tagged Values 646  
Stereotypes 646  
Modeled Extensions 648  
Interoperability using CWM 648  
Adapter Construction 652  
Interoperability Frameworks for CWM 654  
Transform Direct 654  
Transform Indirect 654  
Extending to Web Services 657  
CWM and W3C Standards 657  
CWM Meta Data Interchange Patterns RFP 660  
CWM Web Services RFP 660
The authors wish to acknowledge Sridhar Iyengar for his vision, foresight, and ongoing championing of the CWM effort, ever since the time of its earliest inception. The authors also wish to acknowledge the significant contributions made by both Jean-Jacques Daudenarde and David Last in the development of the CWM model.

Of course, no effort of the magnitude of CWM would ever have been possible without the hard work and contributions of many individuals, and the authors also wish to acknowledge and thank their many colleagues, both within and outside of the Object Management Group, who participated in, contributed materially to, reviewed, and strongly supported, the CWM effort.

And finally, the authors wish to express their gratitude to the fine editorial staff at Wiley Publishing Inc., who recognized the importance of this book from early on, and made its publication possible.
Meta data is widely recognized as the single most important factor in achieving seamless integration and interoperability between dissimilar software products and applications. For software components to interoperate effectively, they must be capable of easily sharing data. And sharing data requires a common definition of how the data is structured (its organization and data types), as well as its meaning (or semantics). Since data is generally defined by meta data, having a common definition of meta data is a necessary prerequisite for achieving integration at the data level. What is required is a common language for describing or expressing meta data and an agreed-upon format or interface for exchanging meta data between components. If both a descriptive language and interchange mechanism for meta data can be standardized and agreed upon by software vendors, then the first and most fundamental roadblock to having truly interoperable systems will have been removed.

The Common Warehouse Metamodel (CWM) is an interoperability standard of the Object Management Group (OMG) that defines a common language and interchange mechanism for meta data in the data warehousing and business analysis domains. CWM provides the long-sought-after common metamodel for describing data warehousing and business analysis meta data, along with an XML-based interchange facility. It has long been acknowledged by leaders and analysts in this particular industry segment that the long-term Return on Investment (ROI) of any complex data warehousing or supply chain effort would be greatly enhanced by the standardization of just such a common metamodel and eXtensible Markup Language (XML) interchange format. CWM enables vendors to build truly
interoperable databases, tools, and applications. Customers benefit by being able to select from best-of-breed product offerings and avoiding single-vendor lock-in, while remaining confident that their investments will not be diluted by the inability of diverse tools to interoperate. CWM has established itself as the meta data interchange standard of choice in the data warehousing and business analysis communities, and has been incorporated into many vendors’ product suites.

From a technical standpoint, CWM extends the OMG’s established metamodeling architecture to include data warehousing and business analysis domain concepts. CWM supports a model-driven approach to meta data interchange, in which formal models representing shared meta data are constructed according to the specifications of the CWM metamodel (essentially an object technology approach to achieving data warehouse integration). These models are stored and interchanged in the form of XML documents. Meta data can be defined independently of any product-specific considerations or formats. It can be stored externally to products as an information commodity within its own right, and is readily used by products as generic definitions of information structures.

Data warehousing and business analysis tools that agree on the fundamental domain concepts and relationships defined by CWM can understand a wide range of models representing particular meta data instances. Tools, products, and applications can integrate at the meta data level, because they have a common language with which to externalize their meta data and do not require knowledge of each other’s proprietary information structures and interfaces. And, although CWM is focused primarily on data warehousing and business analysis, its basic components and methodologies are easily extended to include subject areas of other domains, as well.

**Mission of This Book**

The mission of this book is to provide a comprehensive and highly practical guide for software practitioners who need to implement CWM solutions within their software product offerings, or use CWM-enabled tools in the construction or evolution of their own corporate data warehouses, information factories, and supply chains.

As a developer’s guide to developing CWM-enabled technologies and meta data integration solutions, this book is a particularly novel approach to this subject. In the spirit of Ralph Kimball’s seminal work, *The Data Warehouse Toolkit* (Kimball, 1996), this book approaches the general problem of how to implement CWM by providing four highly representative