IBM PRESS

Enterprise Master Data Management

An SOA Approach to Managing Core Information

Allen Dreibelbis
Ivan Milman
Paul van Run
Eberhard Hechler
Martin Oberhofer
Dan Wolfson

IBM Press
Pearson plc

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco •
New York • Toronto • Montreal • London • Munich • Paris • Madrid •
Capetown • Sydney • Tokyo • Singapore • Mexico City

ibmpressbooks.com

IBM Press Program Managers: Tara Woodman, Ellice Uffer
Cover design: IBM Corporation
Associate Publisher: Greg Wiegand
Marketing Manager: Kourtnaye Sturgeon
Publicist: Heather Fox
Acquisitions Editor: Bernard Goodwin
Development Editor: Songlin Qin
Managing Editor: John Fuller
Designer: Alan Clements
Project Editors: LaraWysong, Elizabeth Ryan
Copy Editor: Bonnie Granat
Proofreader: Linda Begley
Compositor: International Typesetting and Composition
Manufacturing Buyer: Anna Popick

Published by Pearson plc
Publishing as IBM Press

IBM Press offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U.S. Corporate and Government Sales
1-800-382-3419
corpsales@pearsontechgroup.com

For sales outside the U.S., please contact:

International Sales
international@pearsoned.com

This Book Is Safari Enabled

The Safari® Enabled icon on the cover of your favorite technology book means the book is available through Safari Bookshelf. When you buy this book, you get free access to the online edition for 45 days. Safari Bookshelf is an electronic reference library that lets you easily search thousands of technical books, find code samples, download chapters, and access technical information whenever and wherever you need it.

To gain 45-day Safari Enabled access to this book:

- Go to http://www.ibmpressbooks.com/safarienabled
- Complete the brief registration form
- Enter the coupon code 6YCG-LCUE-QIX3-64EQ-NIJ7

If you have difficulty registering on Safari Bookshelf or accessing the online edition, please e-mail customer-service@safaribooksonline.com.

Library of Congress Cataloging-in-Publication Data

Enterprise master data management: an SOA approach to managing core information/Allen Dreibelbis … [et al.]. p. cm.

Includes bibliographical references and index.
ISBN 978-0-13-236625-0 (hardback: alk. paper)
1. Database management. 2. Web services. 3. Computer architecture. I. Dreibelbis, Allen.
QA76.9.D3E68 2008
004.2'2—dc22 2008015422

ISBN-13: 978-0-13-236625-0
ISBN-10: 0-13-236625-8

Text printed in the United States on recycled paper at Courier in Westford, Massachusetts.
First printing, May 2008

Foreword

I meet with senior business and technical executives around the world, in both the public and private sectors, on a daily basis. CEOs, CFOs, CIOs, and line-of-business executives alike are all facing incredible pressures across all fronts. They need to create new shareholder value by improving both the top and bottom lines. They must improve customer service in the face of fast-moving global competition. They must mitigate risks inherent in basic business decision making and avoid fraudulent activities in their own operations. And, as if that isn't pressure enough, they must also deal with a plethora of regulatory requirements.

What they've come to find is that the availability of information provides them with some relief from these almost incessant pressures—the sense of relief that comes from unlocking information and letting it flow rapidly and easily to the people and processes that need it. Trusted information—complete, accurate, timely, insightful information—is delivered in the context of the task at hand.

Take, for instance, one leading electronics manufacturer. By providing unified, timely product master data, the company was able to speed product introduction cycles by weeks and improve the satisfaction of their distribution partners at the same time. An innovative retailer has created an "endless aisle" to drive up in-store sales—even when it doesn't have products on hand. This innovation is enabled with a unified view of product data that spans both the company's own inventory and that of its distributors. In the case of customer master data, a 360-degree view of clients helped one financial services company avoid the risk of offering more credit cards to clients who were already in default with their existing credit

card accounts. Master customer data also helped one telecommunications company capture cross-sell opportunities across its landline, wireless, and long-distance services. The same project was the foundation for improved customer service in the company's call center and helped to reduce customer churn. The possibilities are endless.

Unlocking information and letting it flow rapidly and easily to the people and processes that need it is easier said than done. Over the past 20 years or more, the IT industry has focused on automating business tasks. The result of this effort is a highly complex information landscape; individual automation projects have led to disconnected silos of information. Little trusted information exists—there are multiple versions of the truth. Redundancy reigns—both logically and physically. Few common definitions of key data elements exist or are shared across the enterprise. No common processes for managing and ensuring the integrity of critical data domains exist. These facts define today's environment. They blind the business from the information it needs, add cost to the IT infrastructure, and slow the ability of the business to move forward with confidence.

Solving these problems is what Master Data Management and this book are all about.

As you'll learn, to successfully relieve today's business pressures, Master Data Management (MDM) has to address needs that exist in several distinct but related dimensions. Master Data Management must consider and possibly relate all kinds of Master Data. After all, product data likely relates to some customer data, and perhaps to account data, or perhaps to some other data domain. Unifying these views could lead to more effective customer service. Effective Master Data must also support multiple application styles. Master Data may need to feed an online, transactional ordering system, or perhaps a data warehouse needs Master Data to provide up-sell suggestions to a call center representative. Furthermore, decisions have to be made about how to architect the Master Data Management implementation.

The topic of Master Data Management may seem daunting, but it's really no more daunting than the industry's recent focus on Service-Oriented Architectures (SOA). As a matter of fact, the two topics, SOA and Master Data, are inextricably related. They are two sides of the same coin. A process is only as good as the information it processes, and similarly, information needs to be tied to the context of some process to be of any value. So we must step up to the Master Data challenge. By unlocking the silos of information created by the past 20 years of automation and providing a free flow of trusted information, we will put ourselves in a position to deliver significant value to our organizations.

I encourage you to take advantage of the opportunity this book provides to learn more about Master Data Management. You'll be learning skills you can use to relieve the pressure and deliver more value to your organization. Your time will be well spent. Enjoy the experience.

—Dr. Ambuj Goyal
General Manager, Information Management
of IBM Software Group

Foreword

How does one build a contemporary "super city" that is both technologically forward-looking and compatible with its environment? The challenge is even greater when we build a "super city" that is built on the foundation of an existing metropolis.

Clearly, architecture remains the key challenge in planning enterprise Master Data Management (MDM) infrastructure for the contemporary Global 5000-size enterprise. Experience-based blueprints and architecture patterns are invaluable in such an effort.

In our MDM research with very large-scale enterprises, analysts at the MDM Institute have seen multimillion dollar ($/€/£) projects fail due to poor MDM architectural planning. Such failures included economic failure caused by the inability to cost-effectively scale or political failure caused by the inability to integrate the twenty-first century corporate supply chain.

Inside this highly anticipated book, MDM practitioners will find architectural patterns presented as the nexus of seasoned enterprise architectural experience and early-adopter MDM operational experience. Moreover, the authors have shown their deep experience in delivering an essential guide for every MDM practitioner—from Enterprise Architect to MDM project leadership. This book provides a key technical foundation for understanding the fundamental MDM components and how they work together. As a bonus, the reader will benefit from clear extrapolations on how SOA implementations both benefit from and require MDM.

Enterprise Master Data Management: An SOA Approach to Managing Core Information provides a vital reference architecture for all serious enterprise MDM practitioners.

—Aaron Zornes
Founder and Chief Research Officer,
CDI-MDM Institute

Preface

What Is This Book About?

Master Data Management (MDM) refers to the disciplines, technologies, and solutions that are used to create and maintain consistent and accurate master data for all stakeholders across and beyond the enterprise. *Enterprise Master Data Management: An SOA Approach to Managing Core Information* explains key concepts of MDM, the business value of MDM, and how to architect an Enterprise Master Data Management Solution. The book is a comprehensive guide to architecting a Master Data Management Solution that includes a reference architecture, solution blueprints, architectural principles, and patterns and properties of MDM Systems. The book also describes the relationship between MDM and Service-Oriented Architectures, and the importance of data governance for managing master data. Figure 1 provides a summary of the book's chapters that are summarized in the following list.

Chapter 1: "Introducing Master Data Management" describes the fundamental concepts of master data and MDM. We describe the key characteristics of a Master Data Management System and how the MDM System's ability to manage master data provides benefits to the enterprise. We also introduce the reader to multiple MDM methods and implementation styles.

Chapter 2: "MDM as an SOA Enabler" describes the relationship between MDM and Service-Oriented Architectures. We demonstrate how MDM and SOA work together to help in the achievement of business and IT goals related to managing master data, and explain why we view MDM as an enabler for any SOA-style solution. The chapter includes topics such as SOA concepts, SOA principles, service granularity, service composability, and information services.

Chapter 3: "MDM Reference Architecture" describes the functional characteristics of the Master Data Management Reference Architecture. We describe how to position and design a Master Data Management Solution within an enterprise. We describe

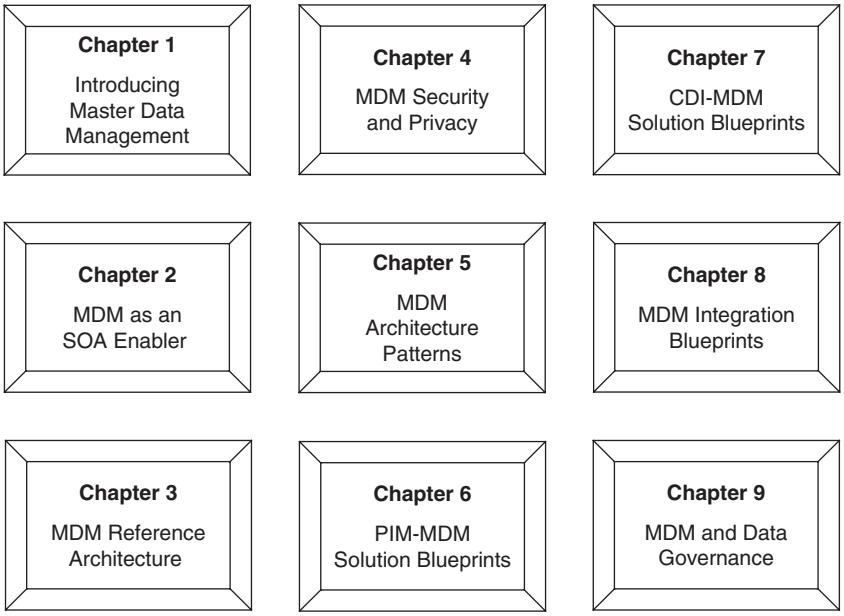

Figure 1 Chapter Summary.

the type of functionality required to deliver a Master Data Management Solution, identify the major architectural building blocks, and then demonstrate how those architectural building blocks collaborate in the delivery of MDM functionality.

Chapter 4: "MDM Security and Privacy" describes the role of security and privacy in an MDM architecture and deployment. We provide insight into developing an understanding of the value of and the risks to master data and then offer guidance for the tasks of selecting and applying the appropriate security controls. We then describe in depth the types of security services that provide the appropriate controls and how those services can apply to the implementation of an MDM Solution.

Chapter 5: "MDM Architecture Patterns" provides an overview of architecture patterns often encountered in MDM deployments. We describe in detail the architecture patterns that helped to shape the MDM Reference Architecture. The architecture patterns encountered were either new architecture patterns, variations of existing architecture patterns, or known architecture patterns that were applied in the area of Master Data Management.

Chapter 6: "PIM-MDM Solution Blueprints" introduces the concept of MDM Solution Blueprints; in this chapter, we explain the relationships between architecture patterns and business patterns for PIM-MDM solutions. The Solution Blueprints are based on the MDM Reference Architecture. Based on specific business requirements for product information management, we describe a variety of PIM-MDM Solution Blueprints for several industries and solution scenarios.

Chapter 7: "CDI-MDM Solution Blueprints" explains the relationships between architecture patterns and business patterns for CDI-MDM solutions. The Solution Blueprints are based on the MDM Reference Architecture. Based on specific business requirements for customer data integration, we describe a variety of CDI-MDM Solution Blueprints for several industries and solution scenarios.

Chapter 8: "MDM Integration Blueprints" provides further guidance on how to integrate an MDM System into an existing IT landscape. We provide guidance and describe sample integration scenarios, such as integrating the MDM System with a Data Warehouse and integrating an MDM System with an SAP application for the authoring of data.

Chapter 9: "MDM and Data Governance" explores the critical nature of data governance in Master Data Management, and how people, process, and technology work together to leverage master data as an enterprise asset. We explore the critical nature of data governance in Master Data Management and the direct and indirect roles that the architecture for the MDM Solution can play in enabling data governance.

Who Should Read This Book

Enterprise Master Data Management: An SOA Approach to Managing Core Information has content that should appeal to a diverse business and technical audience, ranging from the executive level to experienced MDM practitioners and especially those new to the topic of Master Data Management. Newcomers to the topic of MDM, and even SOA, will certainly benefit from the chapters that introduce MDM, that explain security and privacy, and that show how MDM complements the development of a SOA.

Readers with a strong technical background, such as Enterprise Architects, System Architects, and Information Architects, should enjoy reading the detailed content that provides technical guidance for implementing Master Data Management. Technical guidance covers a broad range of topics—implementation styles, methods of use, SOA, security and privacy, architecture patterns, and data governance—which are then all brought together into the Master Data Management Reference Architecture and a set of solution blueprints that span CDI-MDM, PIM-MDM, and MDM Integration Blueprints.

Executives trying to gain an understanding of Master Data Management, and even those who are already in the process of deciding how to proceed, will benefit from the content that introduces Master Data Management, data governance, and the solution blueprints.

What You Will Learn

This book is a comprehensive guide to understanding (1) the importance of Master Data Management, (2) the need for an MDM System, and (3) methods of architecting a Master Data Management Solution. We cover a wide range of topics in our discussions of the use of disciplines, technologies, and solutions to implement Master Data Management. Readers of

this book have the opportunity to increase their knowledge about a broad range of topics related to MDM—from both a business and a technical perspective. Readers will learn the answers to the following questions about Master Data Management, which constitute its core concepts:

- What is Master Data Management, and why is there a need for managed master data?
- How can an MDM System provide a consistent understanding and trust of master data entities?
- What is the relationship between SOA and MDM, and how can MDM enable the implementation of a SOA solution?
- How can a security architecture to maintain the security and privacy of master data be implemented?
- What are the core architectural principles, properties, and patterns for MDM Systems?
- What data governance is critical for the management of master data?

In addition to learning MDM's core concepts, the reader will understand how they are incorporated into the design for a Master Data Management Solution. The MDM Reference Architecture provides the reader with a reference architecture that describes the functional characteristics of an MDM Solution implementation within an enterprise. MDM Integration Blueprints, PIM-MDM and CDI-MDM, and Solution Blueprints then provide the reader with knowledge of how to use the reference architecture and architecture patterns to implement a specific solution to solve a specific set of business problems.

How to Read This Book

There are several ways to read this book. The most obvious way to do it is to read it cover to cover to get a complete end-to-end picture of Enterprise Master Data Management. However, the authors organized the content in such a way that there are four basic reading paths through the book. Figure 2 depicts the following two reading paths:

- To **understand the key concepts of Enterprise Master Data Management**, we suggest reading Chapter 1, Introducing Master Data Management; Chapter 3, MDM Reference Architecture; and Chapter 9, MDM and Data Governance. This path should provide the reader with a clear understanding of Master Data Management, data governance, and how to implement MDM within the enterprise.
- To **understand the key concepts for designing Enterprise Master Data Management Architectures**, we suggest that the reader start with Chapter 1 in order to gain an understanding of the need for an MDM System, methods of such a system's use, and implementation styles. Chapter 2 describes the relationship between MDM and SOA, and how SOA principles can be applied to the design of the solution. Chapter 3 explains the MDM Reference Architecture—the foundation for any MDM deployment—from an architectural perspective. Chapter 4 details aspects of the MDM Reference Architecture that are related to MDM Security and Privacy.

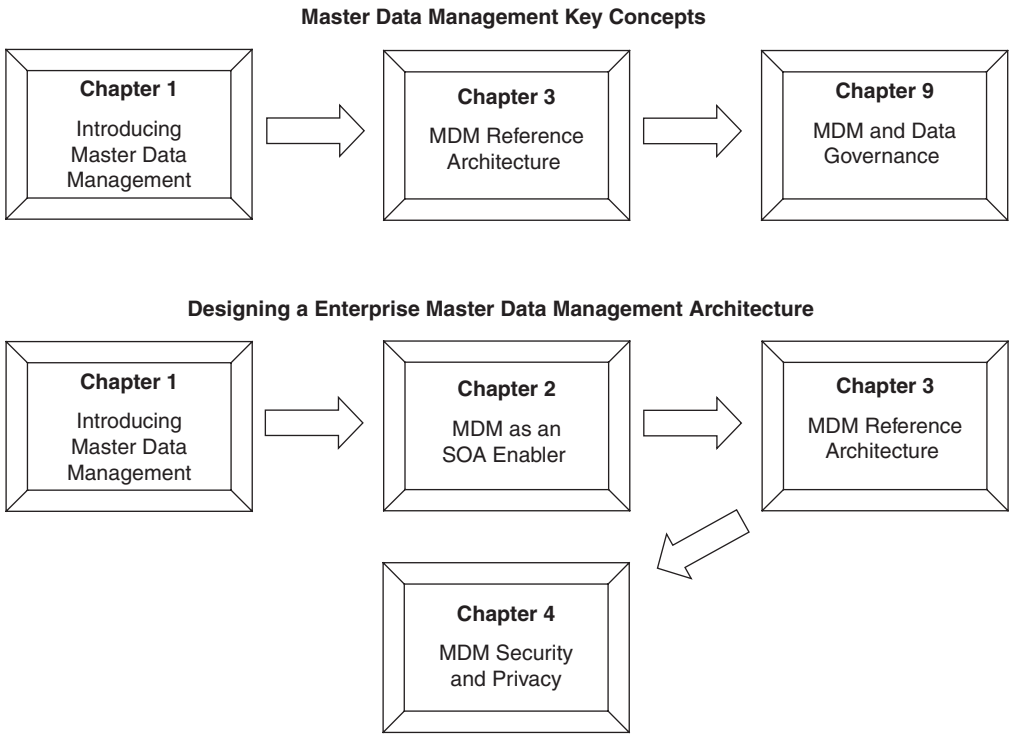

Figure 2 Enterprise Master Data Management Reading Paths.

Figure 3 shows the suggested reading paths for the reader who is interested in detailed knowledge about solution blueprints. In both paths, Chapter 8, MDM Integration Blueprints, is represented as an optional chapter by the use of a dashed arrow and would be of interest to those readers who desire further integration details. Based on the objectives of the reader, Chapter 9, MDM and Data Governance, should be considered for each of the suggested reading paths. The suggested reading paths are as follows:

- **PIM-MDM Solution Blueprints:** If the reader is investigating solutions for the Product master data domain, they are described in Chapter 6. Reading that chapter has prerequisites, though. The reader should be familiar with Chapters 1, 2, and 3. Because the solution architectures are based on the MDM Component Model, a sound understanding of the MDM Reference Architecture is necessary. Explaining certain aspects of a solution also requires an understanding of MDM Security and Privacy, which are described in Chapters 4 and 5. Chapter 5 also explains key areas of the solution architecture.
- **CDI-MDM Solution Blueprints:** Solutions focused on customer master data are described in Chapter 7, which has the same prerequisites as the PIM-MDM Solution Blueprints—namely, Chapters 1, 2, 3, 4, and 5.

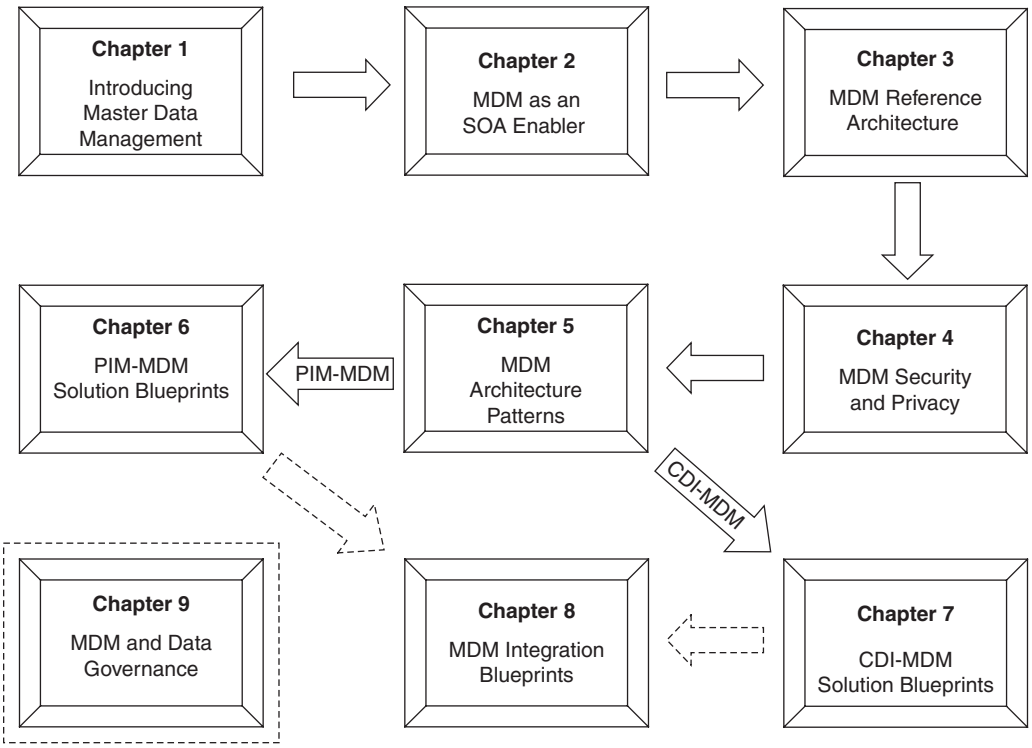

Figure 3 Solution Blueprint Reading Paths.

What Is Not Covered

This book covers many of the key business and technical aspects of Master Data Management and is a comprehensive guide to architecting a Master Data Management Solution, but it does not provide project planning management methodology or an MDM Solution Architecture based on software products. IT Architects will make design decisions and select software products capable of meeting the functional and nonfunctional requirements for a specific solution by considering the existing hardware and software infrastructure for the specific IT environment. Based on our experience, we have provided the reader with a comprehensive guide to designing an MDM Solution. We have also explained the criteria that the reader should use when selecting software for the MDM System. In Appendix B, Software and Solution Offerings for MDM Deployments, we offer a listing of Solution offerings and software products grouped according to the MDM Logical System Architecture. We also provide the reader with the links to relevant Web pages for the providers of those solution offerings.

There are organizational and IT aspects that should be considered as part of the implementation and deployment of an MDM Solution. The people who are developing a project plan

and implementation strategy need to consider the priority of the business problems being addressed, technology gaps, and the current information management capabilities of the organization. We provide guidance throughout the book on how to address some of these aspects, but we have not dedicated any additional chapters to this set of topics. However, the reader will find in Appendix A, MDM User Roles, a summary of the types of user roles, responsibilities, and associated skill descriptions for the various types of roles that we have seen on MDM projects. The primary focus of our efforts has been on providing the reader with the information required for an understanding of the need for Master Data Management and with a comprehensive guide to designing an MDM Solution.

Conventions

Throughout this book, we mention a number of regulations that can influence how to configure, manage, and protect a Master Data Management environment. To help understand those relevant regulations and their scope, we have collected information about each regulation and provided a summary in Appendix C, Master Data Management and Regulations. We include a table that includes other information about each regulation, such as industries affected, master data domain, and Web site addresses where more information about the regulation is available. The book's text has frequent footnote references that direct the reader to Appendix C for further information.

We also refer to a number of industry and technical standards throughout the book and have chosen to include a description of those standards in Appendix D, Standards and Specifications. We do not use footnotes to direct the reader to Appendix D when a standard is mentioned, but we do encourage readers to refer to that appendix if they are unfamiliar with a particular standard. Appendix D contains a table that identifies the standard, provides a link to a Web site for further information, and presents brief notes about the standard.

Many of the chapters have references to additional sources of information that can provide further information about the subject being discussed. External references for each chapter are identified at the end of each chapter and referenced as footnotes. Each reference follows a format that includes the appropriate information relative to the reference, such as the author's name, title of the book or article, date of publication, publisher, ISBN number, and Web site.

Chapter 1

Introducing Master Data Management

As companies struggle to become more agile by implementing information systems that support and facilitate changing business requirements, the management of core information, such as information about customers or products, becomes increasingly important. We call this information **master data.** In many companies, this master data is kept in many overlapping systems and is often of unknown quality. For many organizations, this situation constitutes a dilemma—it becomes increasingly difficult for organizations to implement change. Architectural approaches such as Service-Oriented Architecture (SOA) are difficult to implement when an organization does not have a common definition and management of its core information.

This chapter provides an introduction to master data and the characteristics of Master Data Management (MDM) systems. We start with a brief introduction of master data and explain why the management of master data is important to an enterprise. The second section will outline how different pressures have caused many organizations to lose control of their master data, resulting in a pool of partially redundant, sometimes unmanaged, data. Section three provides a more detailed description of the key dimensions of MDM solutions, describing the different domains of master data, the different ways in which master data management systems are used, and the different ways that MDM Systems can be implemented. The final section discusses how the management of master data can benefit an organization by providing a trusted foundation of authoritative data that can be used consistently across an enterprise and that can also evolve in a managed manner to meet changing business needs.

1.1 Introduction to Master Data Management

The management of key organizational data has always been important. Knowing who your customers are, what products and services you offer, and what the arrangements or accounts you have with your customers and suppliers is fundamental to the operation of most

organizations. Whether your organization is a bank, a retailer, or a government agency, there is a core set of such data that is used across the enterprise. It is used to open new accounts, to introduce new products to the market, and to determine what products to offer customers. This data is called master data.

Master data is some of the most valuable information that a business owns. It represents core information about the business—such as customers, suppliers, products, and accounts—and the relationships between them. Each of these domains of master data represents information that is needed across different business processes, across organizational units, and between operational systems and decision support systems. In essence, master data defines an enterprise.

Master data captures the key things that all parts of an organization must agree on, both in meaning and usage. For example, it is important that all parts of an organization share an understanding of what defines a customer, which customers exist, where customers are located, and what products they have purchased or have been offered. A common understanding is useful both to prevent bad things from inadvertently happening—such as a bill getting posted to the wrong address—and to provide an opportunity for significant business benefits such as improving the ability to sell complementary products to customers. Master data is important in both operational and analytical environments. Many operational business processes touch master data—for example, introducing a new product to the market, signing up a new supplier, and adding a new phone service to a customer account. All of these processes touch many different application systems that must all share a core set of information about products, suppliers, and customers. For the business process to execute properly, this master data must be accurate and consistent. Analytical systems have similar requirements—master data often forms the key dimensions and hierarchies used for reporting and analysis of key business data. Increasingly, analytics are also being applied within operational business processes to better monitor and optimize business transactions. Trustworthy data is a fundamental ingredient of meaningful analytics.

Management of master data is not new. Most organizations have systems to store and retrieve the master data that is critical to their business. Unfortunately, many information systems have become increasingly complex in response to the pressures of growth, business changes, and technology changes. It has therefore become increasingly difficult for organizations to identify, maintain, and use an authoritative set of master data in a consistent way across the enterprise.

Many of the IT architectures[1] that were used to construct existing customer information files or master product databases were designed in support of a few, typically homegrown, applications. More often than not, these applications were themselves constructed in support of a particular line of business rather than for the enterprise as a whole. As the business changed over time, it may have been easier to create a new application and database to handle

1. The term Information Technology (IT) will be used to generically refer to the organization responsible for operating the computing environment in a large organization.

the new requirements than it was to modify all of the existing applications. For example, as the Internet arose as a powerful communications mechanism, companies wanting to enable customer self-service on the Web needed to either extend their existing systems or create new applications and databases to manage this new channel of communication. Because the Web introduced the need to maintain attributes such as e-mail addresses, login names and passwords that existing systems didn't support, many enterprises found it was easier to create new applications and databases to support the Internet channel than it was to extend existing systems. In other words, it was more expedient to build a new system and figure out how to synchronize it with existing systems than it was to extend the existing ones. Although not ideal, doing this may not be a problem if you only have a few simple information systems—but in a larger enterprise, especially one that has grown through multiple mergers and acquisitions, the situation can quickly become problematic. Information Technology (IT) systems, often easier to extend than to change, become increasingly complex with many point-to-point integrations and partially redundant information systems. For example, it is not unusual for a large financial institution to have more than 20 systems that maintain information about customers. Attempting to synchronize customer information across such a large number of systems can result in an unmanageable number of point-to-point integrations—a rat's nest of information flows.

The existence of these partially redundant data stores results in additional complexity throughout the systems architecture of an organization—not to mention additional IT costs. The definitions of what might appear to be common terms, such as "supplier" or "product," will likely be different within the different systems. The data itself may be inconsistent because of systems enforcing different rules for data validation and cleansing. It is therefore difficult for a business to achieve a complete and consistent understanding of master data that is spread across multiple systems if those systems lack the proper controls and integration.

This situation leads to several consequences. Perhaps the most fundamental issue is the quandary for users and applications—where should they go to find and use accurate data? Is there an authoritative source that can be trusted? Without an authoritative source of master data, business processes can become more complex to develop and implement when a complete and accurate view of master data is not available. Similarly, implementing decision support systems can also be tricky without well-known sources for trusted master data.

A second consequence is architectural brittleness—making a small change in one system can have a significant impact on many other systems. Analyzing the scope and cost of a potential change across a complex web of interconnected systems can be difficult. This difficulty is significant, because changes are frequently required to support new business requirements, to support mergers and acquisitions, and to integrate new applications. Indeed, this architectural brittleness significantly impacts the organization's ability to evolve and change according to market pressures. Supporting the growth of a business in terms of operational throughput or geographical distribution yields similar issues. Distributing master data through large clusters of computers requires careful design to manage the synchronization of the master data. The final issue is IT cost—redundant data requires redundant storage as well as the communications and computational infrastructure to maintain it.

The business consequences are perhaps even more significant. When master data is spread across multiple systems in an unmanaged way, there may be multiple competing views of master data: different customer lists, lists of suppliers or product definitions. Without a complete and authoritative set of master information, it is difficult for enterprises to optimize their relationships with customers and suppliers across different product lines; it is difficult to rapidly introduce new products to the market or to relate sales performance to product categories. For example, we have worked with large enterprises where customers in one business unit were suppliers in another—consolidating customers and suppliers into a common Master Data Repository allows the businesses to more effectively negotiate favorable contracts.

In an ideal world, there would be a single place where all common master data in an organization is stored and managed. The data would be accurate, consistent, and maintained in a coherent and secure manner. All updates would take place against this single copy of master data, and all of the different users of master data would interact with this single **authoritative source** of information. For customer data, such an arrangement means that all applications that use customer data would go to a single source; the data stored there would be the customer data used, for example, to open and close accounts across all lines of business, for provisioning accounts, and for all marketing and analysis.

A single source of master data represents three important capabilities: an authoritative source of information, the ability to use the information in a consistent way, and the ability to evolve the master data and the management of the master data to accommodate changing business needs. A suite of services can then be created around this master data that allows the data to be seamlessly integrated into business processes and analytical environments. Together, these three capabilities provide organizations with a powerful foundation for efficient business execution.

An authoritative source of master data can be trusted to be a high-quality collection of data that follows a well-defined and agreed-upon structure. The data has been standardized (e.g., all address information follows the same format). Duplicates have been rationalized. The currency of the information is maintained through continuous or periodic updates. The information is complete, secure, and accurate.

Authoritative master data exposed through reusable business services provides the organization with the flexibility to exploit master data in new ways by enabling applications to follow repeatable, defined patterns of usage for operating over this data. For example, in a banking environment, customer profile information maintained as part of the master customer data can be used consistently across all of the different ways that a customer may interact with the different parts of a business—the branch teller, the call center representative, and the Web—to ensure a smooth and consistent customer experience. Each of the supporting systems would use the same business service to retrieve customer information. As new opportunities to interact with the customer arise—for example, via a customer's cell phone—again, the same services could be once again reused, because both the data and their method of use have been agreed upon.

A critical element in this ideal world of master data is flexibility. As business requirements, regulations, and implementation technologies change, we often find that the definition and

use of the master data needs to evolve along with it. Again, in our ideal world, this type of change would be a nondisruptive change to our environment. For example, if a retailer decides to open up stores in a new country, it should be easy to extend the definition of its products to accommodate additional information, such as new currencies and new regulatory requirements. At the same time, we don't want this simple data model change to break existing applications. The master data environment must thus support a smooth evolution of both the data structures as well as the services that manage the behavior of the master data. In addition to supporting evolution, the ideal system also supports the extension of the environment, for example, to add new domains of master data, and the linkages between them, in a nondisruptive manner.

The goal of **Master Data Management (MDM)** is to enable this ideal world. Through a combination of architecture, technology, and business processes, MDM provides an approach to incrementally reducing the amount of redundantly managed information and providing information consumers throughout an enterprise with authoritative master data. **MDM Systems** that focus exclusively on managing information about customers are often called **Customer Data Integration (CDI)** systems. MDM Systems that focus exclusively on managing the descriptions of products are called **Product Information Management (PIM)** systems. MDM Systems that enable multiple domains of master data, and that support multiple implementation styles and methods of use, are sometimes also called **Multi-Form MDM Systems.** The different technical characteristics of MDM Systems will be described in Section 1.3.

A Master Data Management strategy addresses a wide variety of business and technical concerns within an enterprise. It is often wise to address these concerns incrementally. Incremental deployment allows significant value to be provided as each phase of an MDM project expands the capabilities of the MDM System by integrating additional systems, extending the kinds of data managed, or providing new ways in which the master data may be used. The ideal MDM implementation represents a state of management that organizations may approach over time. An incremental approach allows benefits to be seen and measured throughout the many phases of an MDM implementation project. For these reasons, MDM projects are key strategic initiatives with measurable benefit to the entire enterprise.

MDM is a key facet of a broader enterprise information management strategy. MDM plays a key role within an information architecture as a provider and custodian of master data to the enterprise. This broader strategy must map from the business imperatives to the current and future IT environment. For example, the business need to cross-sell between different lines of business in a large organization may drive the need for MDM. Other drivers, such as the need to comply with Anti-Money Laundering regulations, require banks to apply a broader suite of technologies that are architected in such a way as to detect and respond to potentially fraudulent activities. These examples reveal the broader need to define an enterprise information management strategy and the architectures to support it.

This book provides a comprehensive guide to architecting MDM systems. It includes architectural principles, a reference architecture, patterns, and properties of MDM Systems. We have chosen to approach MDM from the perspective of Service-Oriented Architectures (SOA),

and more specifically, from the viewpoint of treating information as a core set of business services within an enterprise architecture. This approach helps us to address the following three underlying principles—that an MDM System:

- Provides a consistent understanding and trust of master data entities
- Provides mechanisms for consistent use of master data across the organization
- Is designed to accommodate and manage change

Achieving these MDM goals requires MDM software that is capable of both managing master data and providing master data to a community of users and systems. These goals may also lead to organizational change to support the implementation of business practices that properly manage and exploit the master data.

We will explore these principles in further detail throughout this chapter, and they will be woven throughout the rest of the book as a set of underlying assumptions that define the nucleus of an MDM System. In the next sections, we spend more time on why MDM is an important aspect of an enterprise's information architecture. Section 1.3 introduces key technical characteristics of MDM systems, setting the stage for the deeper architectural discussions in subsequent chapters. The final section discusses the business benefits of an MDM system using the three principles described above.

1.2 Why an MDM System?

Master data has been around a long time, so why do we suddenly need MDM Systems to manage this kind of information? What makes this master data special? Why is it important? These are important questions—so within this section we look at how the information that is some of the most valuable information to an enterprise has become virtually unmanaged and often ungoverned. The fundamental purpose of an MDM System is to serve as the authoritative source for master data: An MDM System is a system that provides clean, consistent master data to the enterprise. If the business benefits of a managed master data environment are clear, then why is it that enterprises have unmanaged master data? Why do organizations have multiple, often inconsistent, repositories of data that should be maintained in common across the enterprise? For convenience, we call this data **Unmanaged Master Data.**

First, let's consider the distribution of unmanaged master data throughout a typical enterprise. Why are there multiple copies of master data? Are these copies redundant? This distribution may be viewed along any number of dimensions, including by line of business **(LOB)**, by organizational change—such as mergers and acquisitions, and by the introduction of packaged software.

1.2.1 A Cross-LOB Perspective

Lines of business (LOB) are the natural segmentations of responsibilities that form within an organization, especially where the organization carries a broad portfolio of products or services. By their nature, lines of business often have unique perspectives on core business

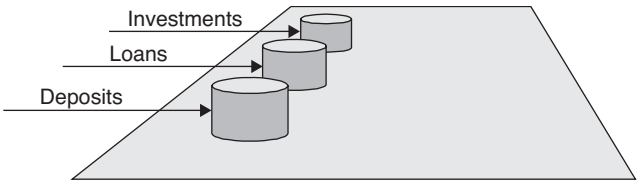

Figure 1.1 Lines of business often maintain their own business information.

information, such as products (the products that this LOB offers), customer (the type of customer information that is important), and account (the nature of an ongoing relationship with a customer based on one or more products). For example, a line of business within a financial institution that is focused on deposits will usually carry different product information than a line of business focusing on investments—see Figure 1.1. Similarly, the customer information that is important to a mortgage department is often different from the information that is important in the support of checking accounts. In reality, neighboring lines of business experience varying levels of cohesion—some lines of business share a great deal of business information, while others share a good deal less.

Within large organizations, lines of business frequently act as very different suborganizations, each funding and maintaining its own suites of applications and data. Even where similar, or indeed identical, applications such as Customer Relationship Management (CRM) or Campaign Management are in use, lines of business often install and manage their own instances of these solutions independently. For example, when multiple LOBs implement a software package like SAP, they often make unique customizations to the data models and look-up tables, which results in multiple independent implementations.

Typically, lines of business capture and maintain unique representations of core business information (customer, product, arrangement), each with their own unique slant on the usage and representation of that information. While regulatory requirements have a role in these differences, in many cases this issue is about control. A line of business sees this data as critical to its day-to-day operations and may not see value in sharing this data within the broader enterprise. All of these factors encourage lines of business to seek to control their own master data, and they sometimes act as barriers to the sharing of this business information.

1.2.2 A Cross-Channel Perspective

Each line of business may also have a number of distinct (distribution) channels to market. While these channels are often very similar, the resulting treatment of business information is often very different. For example, within a single line of business there are frequently entirely different solutions in place for attended channels (such as a branch office) and unattended channels (such as the Internet). These differences in customer interaction patterns across channels often drive a perception that the problem space differs sufficiently to merit an entirely different solution. In other cases, increased complexity is caused by evolution, with emerging channels adopting solutions that were simply not available when support for existing channels was defined.

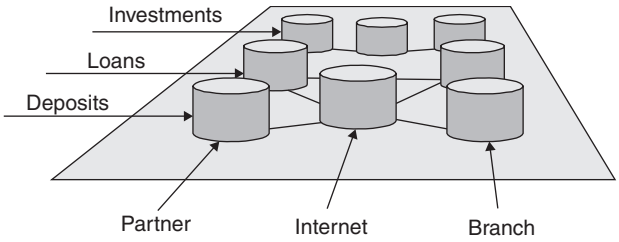

Figure 1.2 Channel variance further scatters business information.

There are, of course, valid differences in business information across channels—Internet-only product offers, in-branch deals, and so on. However, these variances are often best viewed as just that—minor differences of the business information rather than fundamentally different information.

The location of master data can also be influenced by the realization that core information[2] can be shared across lines of business, particularly where the adoption of a new channel acts as a catalyst driving a common view of master data across that channel but not across related channels. The result can be a unification of master data across lines of business for some channels but not for others. For example, many enterprises strive to provide a single point of entry for customer self-service over the Internet. Even if a customer has five different kinds of accounts at a bank (managed by five different systems), the organization will still want to present a unified view of that customer relationship through this channel. Indeed, one of the key business themes that we see across many industries is the desire to change the focus of the business from an account-centric one to a customer-centric one across the enterprise.

All of these factors add an additional dimension to the distribution of master data—the location of master data stores across lines of business, and distribution across channels, as shown in Figure 1.2.

1.2.3 A Cross-Business Subdomain Perspective

Different concerns across lines of business and channels often result in variation, not just in where and how business information is captured, but also in what information is required. In the case of customer data, the branch channel may seek to consider a much broader range of information (customer, contact preferences, contact and case history) than related channels such as partner networks, which may only be interested in a subset of this information. The result can be a fundamentally different scope of business information across channels or lines of business. Each channel and/or line of business may also distribute this information across solutions in different ways, adding further to the complexity of the distribution of customer information, as shown in Figure 1.3.

2. Regulatory requirements may, in some industries, limit the amount of data that can be shared across business segments.

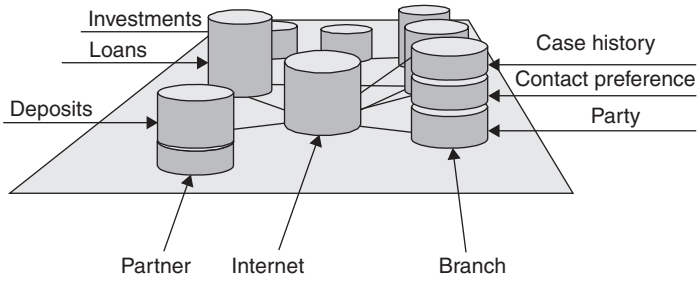

Figure 1.3 Variance in business information further fragments master data.

Additionally, different organization units may have different scopes of interest for product information. Financial controllers, for example, may wish to consider the profitability of different products, product groups, and services. This information is a different subset of product information than the information considered during the process of offering a product to a potential customer.

Variance of the business domain and the information needs of that domain drives further distribution of master data, because each line of business and each channel seeks to maintain its own unique perspective about the business information that best meets its needs.

1.2.4 A Cross-Application/Technology Perspective

An increasingly common reason for master data redundancy is the introduction of packaged applications and solutions (CRM, ERP, etc.). Typically, packaged systems are designed to manage their own master data. When multiple packaged applications are deployed in an environment, an interesting conundrum arises—each of the packaged applications will likely only store the information it needs for its own operations—so when you have two or more of them deployed in an environment, there is no common definition of the master data elements. For example, information about customers is normally needed by both an ERP system and a CRM system—because they each likely maintain unique customer attributes, neither represents a complete view of the customer information. As we describe in Chapters 3 and 5, a common pattern is to use the MDM System to support the complete representation of customer information through the aggregation or federation of customer data from multiple systems.

Integrating a packaged application system into an enterprise can be a difficult and costly endeavor, because the new system must be synchronized with existing sources of data, including master data. When packaged solutions contain multiple independent applications, they may need to synchronize master data within their own solutions as well as with the customer's environment. In an environment with many such applications, pair-wise synchronization can be complex and fragile. Using an MDM System as a common hub from which other systems are synchronized simplifies the number of connections and can improve the overall quality of master data and the manageability of the environment.

Finally, consider the effect of variance in the technical platform or application solution on the distribution of master data. Between lines of business or channels, many different representations of business information may evolve based on different platforms or applications. For example, if a customer has a well-tuned mainframe application already managing product data, extending that system to supply information to a new channel application may be perceived as too costly. Real or not, the perception is often that these platform differences are difficult to resolve, and often no attempt is made to integrate across different systems, which results in further distribution of unmanaged master data.

The result of all of these varying concerns is that master data is often widely scattered across the enterprise, with each channel, line of business, and solution stack evolving its own unique silo of master data. Where attempts to share business information do exist, they are usually ad hoc in nature and limited to a particular channel or product type. For example, it is not uncommon to find at least a couple of dozen stores of customer data in a financial institution.

1.2.5 Mergers and Acquisitions

Mergers and acquisitions serve to dramatically accelerate the replication of business information within an enterprise. Each party to a merger has its own distinct set of master data sources along the dimensions highlighted earlier—a sort of master data fingerprint. Without extensive effort to converge these data stores, the resulting merged organization will not be able to effectively leverage the combined assets (customers, products, etc.) of the new organization or be able to achieve economies of scale in the operation of the merged enterprise. For example, consider a case where organization A is to merge with organization B. Both organizations maintain LOB-specific stores of master data, as shown in Figure 1.4; however,

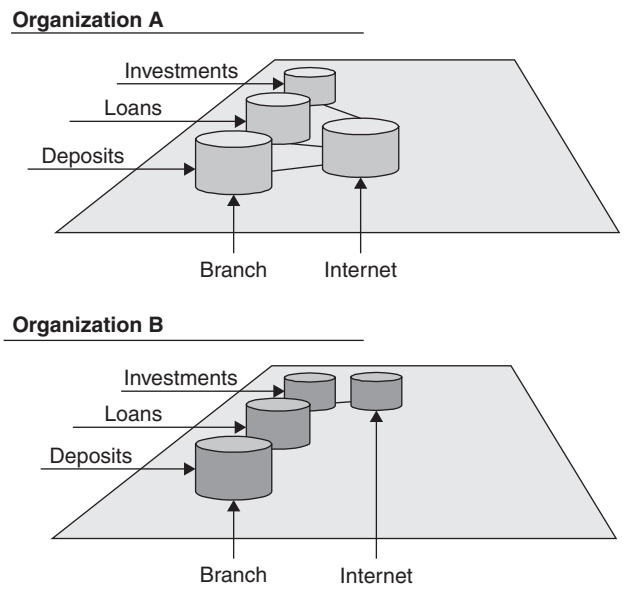

Figure 1.4 Two organizations with different patterns of data distribution.

both organizations consider themselves to have developed strong offerings for the Internet channel. Organization A shares information across lines of business within the Internet channel, while organization B has all but eliminated channel-specific perspectives on master data, and each line of business operates on the same data, regardless of the channel concerned.

Merging these organizations yields a very different picture, however. The result is a dramatic increase in line of business-specific solutions, because each organization brings to the table its own solution, in each line of business. Within a specific channel (e.g., Internet), the two contrasting approaches of sharing information across lines of business, and eliminating channel-specific variances do not align well, which results in further complexity, as shown in Figure 1.5, with one solution seeking to be a channel-specific source of truth for all lines of business, and another seeking to eliminate channel-specific management of business information.

Consolidation and modernization of existing systems often require a similar kind of convergence. For example, an organization may, after several years of geographic growth, realize that each region has independently created localized systems that contain partially replicated and overlapping sets of data, which has led to an incomplete and inconsistent view of its customers, suppliers, and products. Addressing these business problems can be viewed as a merger of the different geographically based organizations and systems.

In summary, there are many natural forces that have led many enterprises to have multiple copies of master data spread out across different lines of business, different communications channels, and different kinds of applications. As the number of unique copies of master data increases, synchronization via point-to-point connections becomes more complex, and the overall environment becomes more difficult to both manage and change. Indeed, when multiple systems manage the metadata, it is hard to achieve consistency of the information. For example, it is likely that many of the applications used to manage master data will have different rules for validating and standardizing the data—thus, even simple things like shipping address information for a customer may not be consistent.

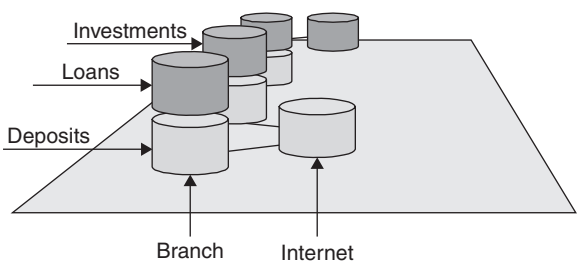

Figure 1.5 The resulting merger often suffers from the worst of all inputs.

1.3 What Is a Master Data Management System?

Master Data Management Systems provide authoritative data to an organization. But what kind of data? How do we work with the MDM System? How do we integrate the MDM System with the existing systems? These questions describe a solution space within which there are a wide variety of ways in which MDM Systems can be deployed and used according to the needs and goals of the enterprise.

In this section, we describe the three primary dimensions of this MDM solutions space. As shown in Figure 1.6, the three dimensions are the **domains of master data** that are managed, the **methods** by which the system is to be used, and the **styles of implementation** that are needed for a particular deployment. It is important to note that MDM implementations are typically not deployed in a "big bang" approach where all domains are managed across all methods of use. Organizations generally start with a limited scope that provides the highest return on investment in a relatively short time frame. As MDM implementations are rolled out over several phases, the space of the implementation may grow. Additional domains are added, the method of use may expand, or the implementation style may change to deliver additional business value. The term **Multiform MDM** is sometimes used to describe MDM Systems that support these three dimensions of MDM Systems. The following sections describe these dimensions in greater detail.

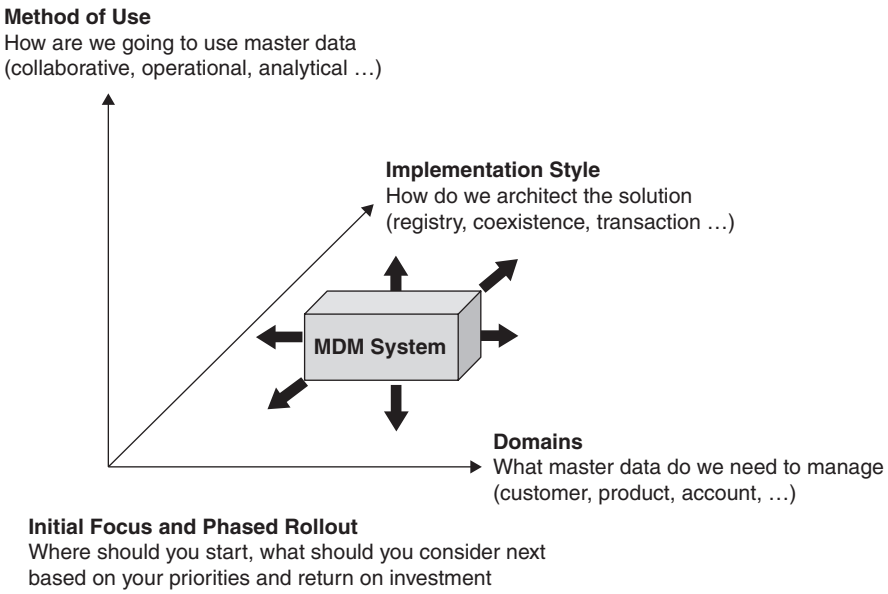

Figure 1.6 Dimensions of Master Data Management.

1.3.1 Master Data Domains

Master Data Management has emerged over the last few years from the recognition that the existing markets of Customer Data Integration (CDI) and Product Information Management (PIM) had key similarities as well as differences. CDI focuses on managing people and organizations—which we will collectively call **parties.** A CDI system can aggregate party information from many preexisting systems, manage the use of the party data, and distribute the information out to downstream systems such as billing systems, campaign management systems, or CRM systems.

PIM systems manage the definition and lifecycle of a finished good or service—collecting product information from multiple sources, getting agreement on the definition of products, and then publishing this information to Web sites, marketing systems, merchandizing systems, and so on. PIM systems are distinct from Product Lifecycle Management (PLM) systems, which focus on the design and development of products rather than the preparation of product information to support sales and distribution. There is a natural flow of information from a PLM system to a PIM system as a product transitions from engineering into marketing and sales.

CDI and PIM both represent a common pattern—that of aggregating data from existing systems, cleaning and augmenting that data, and then distributing that data to downstream systems. PIM and CDI systems differ in the most common ways in which the data is used after it has been loaded into the MDM System—we discuss the different methods of use in the following section. It is important to note that MDM Systems do more than just store and retrieve data—they incorporate business logic to reflect the proper management and handling of the master data. The rules for handling a product lifecycle are different than those for managing the lifecycle for a customer. The MDM System may also be configured to issue alerts when interesting things happen. For example, billing systems may need to get notified immediately when a customer address changes. This business logic can be customized for a particular deployment to reflect the needs of a particular industry as well as the unique characteristics of the implementing organization.

As CDI and PIM products have matured, it was also observed that while CDI systems focused on the customer, it was often convenient for such systems to include references to the products or accounts that a customer has. Similarly, PIM systems often need to store or reference the suppliers of the products or services. Supporting and using these cross-domain relationships has become a significant aspect of MDM Systems.

The kinds of information treated as master data varies from industry to industry and from one organization to another. An insurance company may wish to treat information about customers, policies, and accounts as master data, while a telecommunications company may be concerned with customers, accounts, location (of cell phone towers), and services. A manufacturer may be focused on managing suppliers, customers, distributors, and products. A government agency may want to focus only on citizens and non-citizens. In these examples, we see a lot of commonality as well as differences. In general, master data can be categorized according to the kinds of questions they address; three of the most common questions—"Who?," "What?," and "How?" are addressed by the **party, product,** and **account**

domains of master data. Each of these domains represents a class of things—for example, the party domain can represent any kind of person or organization, including customers, suppliers, employees, citizens, distributors, and organizations. Each of these kinds of party shares a common set of attributes—such as the name of the party, where it is located (a party may have multiple locations such as home, work, vacation home, etc.), how to contact it, what kind of relationship the organization has with the party, and so forth. Similarly, the product domain can represent all kinds of things that you sell or use—from tangible consumer goods to service products such as mortgages, telephone services, or insurance policies. The account domain describes how a party is related to a product or service that the organization offers. What are the relations of the parties to this account, and who owns the account? Which accounts are used for which products? What are the terms and conditions associated with the products and the accounts? And how are products bundled?

Location information is often associated with one of the other domains. When we talk about where a product is sold, where a customer lives, and the address at which an insurance policy is in effect, we are referring to location information. Location information is tied to a product, a party, or an account—it does not have an independent existence. There are, of course, cases where location does exist independently, but those situations seem to be less common. Another interesting facet of location is that it can be described in many different ways (by postal address, by latitude and longitude, by geopolitical boundaries)—we need a particular context in order to define what we mean. A location can be a sales territory, a city, a campus with many buildings, a store, or even a spot on a shelf in an aisle within a store. For these reasons, we will treat location as a subordinate domain of master data.

Figure 1.7 shows how the three primary domains of party, product, and account overlap. These areas of overlap are particularly interesting, because they indicate fundamental

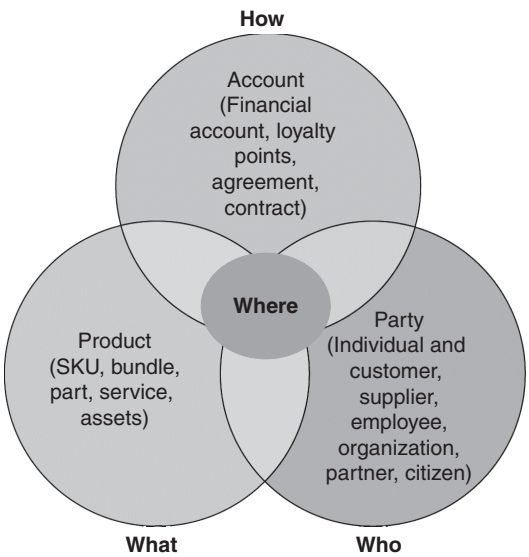

Figure 1.7 Domains of Master Data.

relationships between the domains. For example, when we define a product, we often need to specify the party that supplies that product and the location(s) in which the product may be sold. Explicitly capturing these relationships within the same environment allows us to address business questions that may be otherwise difficult to resolve. Building on the previous example, if we record the party that supplies a product as well as the parties that we sell products to, then we can determine which of our suppliers are also our customers. Understanding the full set of linkages that an organization has with a partner can be valuable in all aspects of working with that partner—from establishing mutually beneficial agreements to ensuring an appropriate level of support. Indeed, perhaps the key benefit of supporting multiple domains of master data within the same system is that it clarifies these cross-domain relationships.

Master data domains can be made specific to a particular industry through the application of industry standards or widely accepted industry models.[3] Typically, standards and models can be used to drive not just the definition of the data model within an MDM Solution but the services that work with the master data as well. In particular, use of standards and models aligns the services exposed by an MDM Solution with accepted industry-specific definitions, which reduces the cost of integration.

Gaining agreement on the definition of an MDM domain can be challenging when different stakeholders within an organization have different requirements or look at the same requirements from different points of view. If well-accepted industry models or standards exist, they can serve as a foundation for further customization, eliminating the need to laboriously gain agreement on every term or service definition. Table 1.1 provides a list of some of the standards and models that are available within a range of industries. Some of these standards and models could be used to guide the definition of data structures and access services for MDM domains.[4]

In summary, an MDM System supports one or more domains of master data. The domains provided are often industry-neutral but can be subsequently tailored (and/or mapped) to different industry standards or models. The domain definitions can be further customized during the design and implementation of an MDM Solution for a specific environment.

1.3.2 Methods of Use

As we look at the roles that master data plays within an organization, we find three key methods or patterns of use: **Collaborative Authoring**, **Operational**, and **Analytical**, shown in Figure 1.8. The simplest way to think about these methods of use is to consider who will be the primary consumers of the master data. Under the Collaborative Authoring[5] pattern,

3. Examples of industry models can be found in Appendix B.

4. The Solution Blueprints described in Chapters 6, 7, and 8 describe how several of these standards and models can be leveraged within an MDM Solution.

5. *Note:* We will sometimes just use the term Collaborative to mean Collaborative Authoring.

Table 1.1 Some Industry Standards and Models

Industry	Standard or Industry Model	Web Resource
Banking	IBM Information FrameWork (IFW)	www-306.ibm.com/software/data/ips/products/industrymodels/
	Interactive Financial eXchange (IFX)	www.ifxforum.org
Insurance	IBM Insurance Application Architecture (IAA)	www-306.ibm.com/software/data/ips/products/industrymodels/
	Association for Cooperative Operations Research and Development (ACORD)	www.acord.org
Telecoms	Shared Information/Data Model (SID)	www.tmforum.org
	IBM Telecommunications Data Warehouse	http://www-306.ibm.com/software/data/ips/products/industrymodels/telecomm.html
Retail	Association for Retail Technology Standards (ARTS)	www.nrf-arts.org
	IBM Retail Data Warehouse	http://www-306.ibm.com/software/data/ips/products/industrymodels/retail.html
Healthcare	Health Level 7 (HL7)	www.hl7.org

the MDM System coordinates a group of users and systems in order to reach agreement on a set of master data. Under the Operational pattern, the MDM System participates in the operational transactions and business processes of the enterprise, interacting with other application systems and people. Finally, under the Analytical pattern, the MDM System is a source of authoritative information for downstream analytical systems, and sometimes is a source of insight itself.

A particular element of master data such as a product or an account may be initially authored using a collaborative style, managed operationally through the operational style, and then published to other operational and analytical systems. Because MDM Systems may be optimized to one or more of the methods of use, more than one MDM System may be needed to support the full breadth of usage. Where multiple MDM Systems are used to support multiple usage patterns, careful attention to the integration, management, and governance of the combined system is required to ensure that the master data of the combined system is consistent and authoritative.

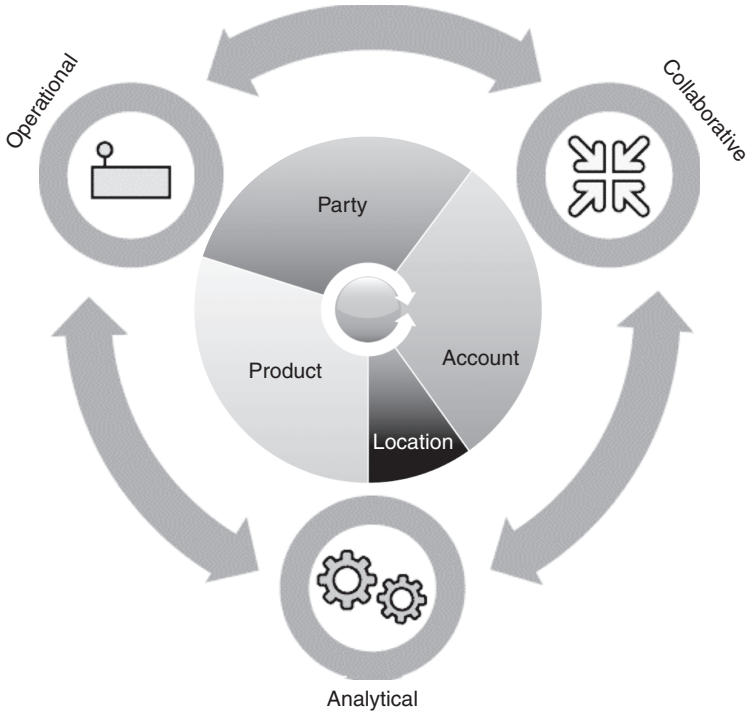

Figure 1.8 Multiple MDM domains and multiple methods of use.

It is important to note that the style of usage is completely independent from the domain of information managed. Although Product Information Management systems are often associated with a Collaborative Authoring style of use, and Customer Data Integration systems are often associated with an Operational usage style, this alignment is not necessary or exclusive. There are an increasing number of cases where organizations seek an operational usage of product information as well as a range of use cases for collaborative authoring of customer information.

1.3.2.1 Collaborative MDM

Collaborative MDM deals with the processes supporting collaborative authoring of master data, including the creation, definition, augmentation, and approval of master data. Collaborative MDM is about achieving agreement on a complex topic among a group of people. The process of getting to agreement is often encapsulated in a workflow that may incorporate both automated and manual tasks, both of which are supported by collaborative capabilities. Information about the master data being processed is passed from task to task within the workflow and is governed throughout its lifecycle.

As a consequence of the complexity of product development and management, PIM systems commonly support a collaborative style of usage. Perhaps the most common process

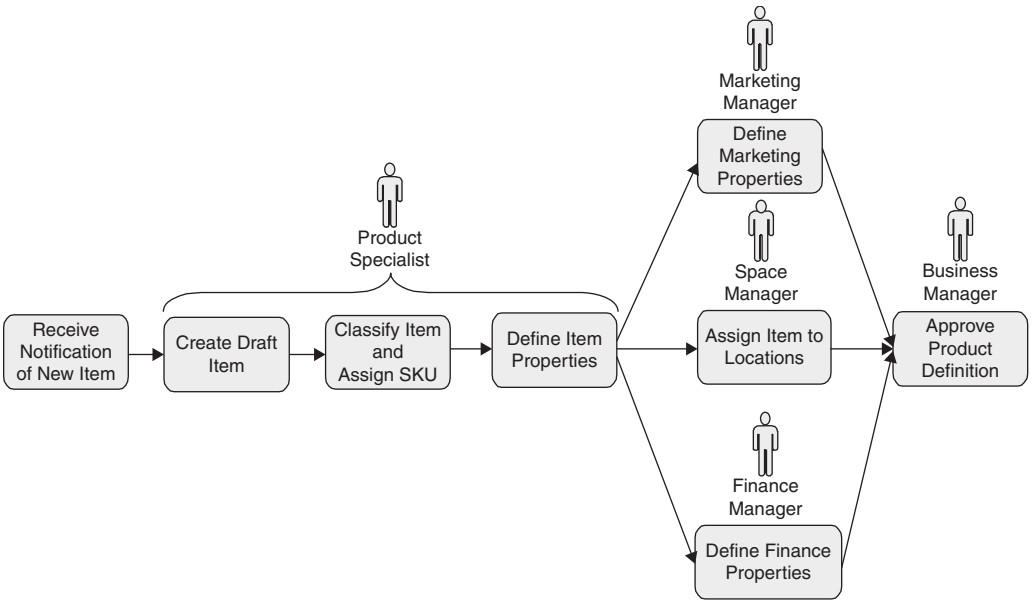

Figure 1.9 Simplified New Product Introduction process.

implemented by PIM systems is the process for introducing a new product to the market. An in-depth discussion on NPI can be found in Chapter 6. A typical NPI process is shown in Figure 1.9.

Here we can see that information about new products (or items) is received from one or more external sources and then incrementally extended, augmented, validated, and approved by a number of different end users with different user roles and responsibilities.[6]

The collaborative steps within a New Product Introduction process are used to define the kinds of properties that describe the product. A given product will be described by dozens, and often hundreds, of properties depending on how the product is classified and where it is sold. In the New Product Introduction process, product specialists, buyers, and other stakeholders describe all of the characteristics of the product that are necessary to bring it to market. These characteristics may include product specifications, marketing information, ingredients, safety information, recycling information, cost, and so on. Large retailers may have more than a million products that they sell, spanning categories from food to clothing to furniture to appliances. The kinds of properties that are relevant to a product depend on the kind of product it is. For clothing, examples include color, size, and material; for electronic appliances, examples might be specifications, color, warranty, and so on. The Collaborative MDM System helps users to capture all of the different relevant properties of the product,

6. Appendix A provides an in-depth discussion of the user roles in an MDM environment.

validate the properties, categorize the product, and coordinate the approval of the product. As buyers and product specialists come up with new ways to describe products, new properties are created to hold these new descriptions. In retail environments, the structure of the product information is constantly evolving.

Collaboration is a common pattern and can be found beyond the PIM domain. Indeed, we find that many of the tasks performed by a product specialist in the PIM environment are also performed in the management of Customer and Account information. A key role that spans all domains of master data is that of **data steward.** A data steward looks after the quality and management of the data. For example, when we believe that two or more party records in a data store may really refer to the same individual, data stewards may need to manually combine information from the party records together and then validate the proposed changes with supervisors. Similarly, where questions requiring human intervention arise about the accuracy of information, a request for attention may be made visible to all data stewards who are capable of handling the issue, which can result in a collaborative pattern to resolve data quality issues.

The Collaborative style of usage requires a core set of capabilities within the MDM environment. A combination of workflow, task management, and state management are needed to guide and coordinate the collaborative tasks and the master data being collaborating on. Workflow controls the execution of a sequence of tasks by people and automated processes. Task management prioritizes and displays pending work for individuals to perform, while state management helps us to model and then enforce the lifecycle of the master data.

Because many concurrent users and workflows may be executing in parallel, the integrity of the master data needs to be protected with a check-in/check-out or similar locking technique. To improve efficiency, master data records are often processed in batches within the same workflow, which results in the concept of a "workbasket" of master data records that is passed from task to task within the workflow. Tasks within a workflow may be automated actions (such as import, export, or data validation) or manual tasks that allow users to work directly with the master data. Typically, this workflow will involve business users and data stewards, a process that, in turn, has implications for the design of the UIs (user interfaces) for collaborative authoring of master data. User interfaces must be both efficient and comfortable to use, and must rely on a set of underlying services that create, query, update, and delete the master data itself, the relationships between the master data, and other related information, such as lookup tables. Tooling to support the flexible creation and customization of collaborative workflows and even user screens may also be provided.

Finally, a common set of services are typically also provided to enforce security and privacy, and to support administration, validation, and import/export of master data. These services are needed across all kinds of MDM Systems.

1.3.2.2 Operational MDM

In the Operational style of MDM, the MDM server acts as an Online-Transaction Processing (OLTP) system that responds to requests from multiple applications and users. Operational MDM

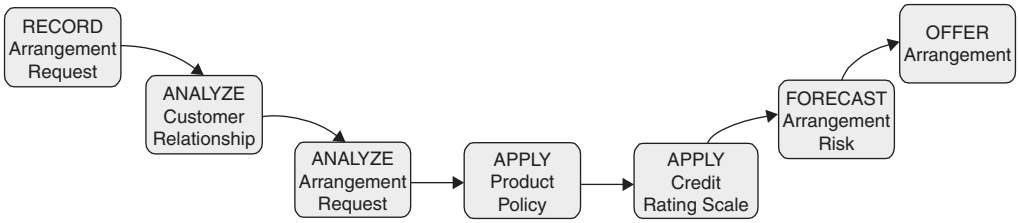

Figure 1.10 Example New Account Opening process.

focuses on providing stateless services in a high-performance environment. These stateless services can be invoked from an enterprise business process or directly from a business application or user interface. Operational MDM services are often designed to fit within a Service-Oriented Architecture as well as in traditional environments. Integration of an Operational MDM System with existing systems calls for the support of a wide variety of communications styles and protocols, including synchronous and asynchronous styles, global transactions, and one-way communications.

A good example of Operational MDM usage is a New Account Opening business process. In this process, a person or organization wants to open a new account—perhaps a bank account, a cable TV account, or any other kind of account. As shown in Figure 1.10, MDM services are invoked to check what information about the customer is already known and to determine if product policy is being complied with before an offer of a new account is made. If the customer isn't already known, then the new customer is added to the MDM System and a new account is created (presuming that the new customer meets the appropriate requirements). Each of the tasks within this workflow is implemented by a service, and many of these services are implemented by an Operational MDM System.

Operational MDM is also commonly used in the PIM domain. For retailers, after products have been defined, the approved product information may be published to an operational MDM System that then serves as a hub of MDM information that interacts with merchandising, distribution, or e-commerce applications. As such applications become more open and able to interact within an SOA environment, the need for such an operational MDM hub increases.

A wide range of capabilities is required for the Operational usage style. There can be hundreds of services that provide access and management of MDM data. Specific sets of services for each kind of MDM object managed provide for creation, reading, updating, and deletion of the MDM objects. Services are also provided to relate, group, and organize MDM objects. As with the Collaborative style of MDM, services are also needed for cleansing and validation of the data, for detection and processing of duplicates, and for managing the security and privacy of the information.

1.3.2.3 Analytical MDM

Analytical MDM is about the intersection between **Business Intelligence (BI)** and MDM. BI is a broad field that includes business reporting, data warehouses, data marts, data mining, scoring, and many other fields. To be useful, all forms of BI require meaningful, trusted data. Increasingly, analytical systems are also transitioning from purely decision support to more operational involvement. As BI systems have begun to take on this broader role, the relationship between MDM Systems and Analytical systems has also begun to change.

There are three primary intersections between MDM and BI.

- **MDM as a trusted data source:** A key role of an MDM System is to be a provider of clean and consistent data to BI systems.
- **Analytics on MDM data:** MDM Systems themselves may integrate reporting and analytics in support of providing insight over the data managed within the MDM System.
- **Analytics as a key function of an MDM System:** Specialized kinds of analytics, such as **identity resolution**, may be a key feature of some MDM Systems.

One of the common drivers for clean and consistent master data is the need to improve the quality of decision making. Using an MDM System to feed downstream BI systems is an important and common pattern. The data that drives a BI system must be of a high quality if the results of the analytical processing are to be trusted. For this reason, MDM Systems are often a key source of information to data warehouses, data marts, Online Analytical Processing (OLAP) cubes, and other BI structures. The common data models for data warehouses use what are called **star schemas** or **snowflake schemas** to represent the relationship between the facts to be analyzed and the dimensions by which the analysis is done.[7] For example, a business analyst in a retail environment would be interested in understanding the number or value of sales by product or perhaps by manufacturer. Here, the sales transaction data is stored in fact tables. Product and manufacturer represent dimensions of the analysis. We can observe that often master data domains align with dimensions within an analytical environment, which makes the MDM System a natural source of data for BI systems.

The insight gained from a data warehouse or OLAP cube may also be fed back into the MDM System. For example, in the travel and entertainment industry, some companies build analytical models that can project the likely net lifetime revenue potential of a customer. To build these projections, they will source the master data from an MDM System and transactional details from other systems. After the revenue potential is computed, the MDM System is updated to reflect this information, which may now be used as part of each customer's profile. Reservation systems can then use this profile to tailor offers specifically to each customer.[8]

Insight may also be derived from data maintained by the MDM System itself. An MDM System contains all of the information needed to report on key performance indicators such

7. An overview of Data Warehousing can be found in [1].
8. More details on MDM and Data Warehouse integration may be found in Chapters 5 and 8.

as the number of new customers per week, the number of new accounts per day, or the average time to introduce a new product. Reporting and dashboarding tools can operate directly over the master data to provide these kinds of domain-specific insights. Some MDM Systems also incorporate a combination of rules and event subsystems that allow interesting events to be detected and actions to be taken based on these events as they happen. For example, if a customer changes addresses five times in three months, that may trigger an alert that notifies event subscribers to contact the customer to validate his or her address on a periodic basis. Analytics may also be executed as an MDM transaction is taking place, using architected integration points that allow external functions to be invoked as part of an MDM service. A good example is the use of scoring functions to predict the likelihood of a customer canceling accounts at an institution. Such scoring functions can be developed by gaining a deep understanding of an issue, such as customer retention, through data mining and building a model of recurring customer retention patterns based on the combination of customer and transaction data maintained within a data warehouse. While it is time-consuming to develop and validate such a model, the scoring model that results can be efficiently executed as part of an MDM service. This kind of analytics is called **in-line analytics** or **operational analytics** and is an important new way in which MDM Systems can work together with BI systems to provide additional value to an enterprise.

The final kind of MDM analytics is where the MDM System provides some key analytic capabilities. One particular kind of insight that can be derived from the information within an MDM System is the discovery of both obvious and non-obvious relationships between the master entities managed. An obvious kind of relationship would be one that discovered households based on a set of rules around names, addresses, and other common information. A non-obvious kind of relationship might find relationships between people or organizations by looking for shared fragments of information, such as a common phone number, in an effort to determine that people may be roommates. Searching for non-obvious relationships may also require rules that look for combinations of potentially obfuscated information—for example, transposed Social Security numbers and phone numbers—to identify potential relationships where people may be trying to hide their identities. Identity resolution and relationship discovery are important for both looking for questionable dealings[9] and understanding a social network that a person is part of—and therefore are important for predicting the overall value of a person's influence.

The analytical style of usage encompasses a variety of capabilities. Populating external analytical environments such as data warehouses with data from an MDM System requires information integration tools to efficiently transfer and transform information from the MDM System into the star or snowflake schemas needed by the data warehouse. Integration with reporting tools is required in order to display key performance indicators and how they change over time. Rules, scoring, and event management are important capabilities for in-line analytics within the MDM environment.

9. More details on MDM-Analytical integration can be found in Chapters 3, 5, and 8. A blueprint describing the use of MDM in threat and fraud scenarios can be found in Chapter 7.

In practice, MDM usage will often cross the boundaries between collaborative, operational, and analytical usage. For example, collaborative MDM processes can be very useful in managing the augmentation of complex operational structures such as organizational hierarchies. On the other hand, there is valuable analytical information that can be gathered around the nature of the collaborative processes. An MDM implementation may start with the usage style that is most important to achieving their business need and then later extend the environment to incorporate additional styles to meet further requirements.

1.3.3 System of Record vs. System of Reference

The goal of an MDM System is to provide authoritative master data to an enterprise. Ideally, a single copy of key master data would be managed by an MDM System—all applications that needed master data would be serviced by this system, and all updates to master data would be made through the MDM System. The master data in the ideal MDM implementation can be considered a **system of record.** That is, the data managed by the MDM System is the best source of truth. If applications want to be sure that they are getting the most current and highest-quality information, then they consult this source of truth. Achieving this ideal MDM System can be difficult, at best, due to several confounding factors, such as:

- The complexity and investment in the existing IT environment
- Master data locked into packaged applications
- Requirements for performance, availability, and scalability in a complex and geographically distributed world
- Legal constraints that limit the movement of data across geopolitical boundaries

All of these factors contribute to the need for copies of master data—sometimes partial subsets, sometimes completely redundant replicas. These copies can be well-managed, integrated, and synchronized extensions of an MDM System. When the replica of the master data is known to be synchronized with the system of record—in a managed way that maintains the quality and integrity of the data within both the replica and the system of record—we can call this copy a **system of reference.** Although it is synchronized with the system of record, it may not always be completely current. Changes to the system of record are often batched together and then applied to the systems of reference on a periodic basis. In some cases, the copy may represent a special-purpose MDM implementation that has been specifically tuned to the needs of a particular style of usage. A system of reference is a source of authoritative master data because it is managed and synchronized with the system of record. It can therefore be used as a trusted source of master data by other applications and other systems.

An MDM system of reference is best used as a read-only source of information, with all of the updates going through the system of record. Figure 1.11 illustrates a simple environment where the system of record aggregates master information from multiple sources, is responsible for cleansing and managing the data, and then provides this data to both managed systems of reference as well as directly to other consumers of the managed information. We use the terms **Managed** and **Unmanaged** to define the scope of the overall MDM environment. In a managed environment, each source should only feed one system, and each consumer should receive information from only one system. Within the managed environment, we can track the movement of information between systems and audit the transactions that use the system.

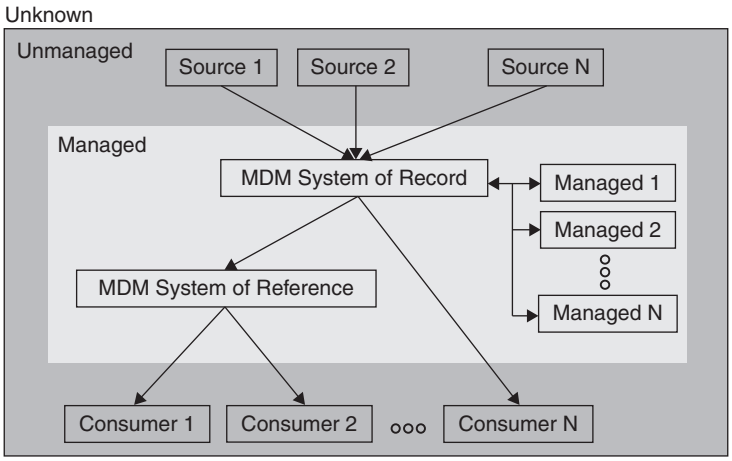

Figure 1.11 System of Record vs. System of Reference.

It is important to note that the system of record may, in fact, be made up of multiple physical subsystems—but this arrangement should be transparent to all of the other systems and people that interact with it. For example, if legal requirements in a country dictate that personal information may not leave the country, then different MDM Systems may be required. In this situation, the different MDM Systems can be logically brought together through the technique of **federation.**[10] The fact that there is more than one MDM System can be hidden from the consumers. Consumers issue requests to the MDM System as a whole and receive a response without having to know (or care) which particular system responded.

1.3.4 Consistency of Data

When data is replicated in an environment, questions of consistency among the replicas immediately arise. For example, as we discussed in the previous section, a system of reference may not always be completely consistent with a system of record. It is useful to think about two basic approaches towards consistency. The first we can call **Absolute Consistency,** and the second we can call **Convergent Consistency.** In a distributed system with absolute consistency, information will be identical among all replicas at all times that the systems are available (for simplicity, let's ignore the case where a system is recovering after a failure). In a distributed environment, we commonly achieve absolute consistency by following a two-phase commit transaction protocol that is provided in most distributed databases, messaging systems, and transaction systems. The basic idea is that we can define a unit of work containing several actions that must either all complete or all fail. We can use this approach to write applications that update multiple databases and guarantee that either all of the

10. The technique of federation brings data from different systems at need. A discussion of federation at the database level can be found in [2].

updates worked or they all failed. We can also use this approach to concurrently update master data records in multiple repositories simultaneously, and with the two-phase commit transaction protocol, either all of these databases will be updated or none of them will be. In either case, all of the databases contain the same information and are therefore absolutely consistent. Two-phase commit can be costly from a performance, complexity, and availability point of view. For example, if we have three systems that all normally participate in a transaction, all three of the systems must be available for the transaction to complete successfully—in other words, if any one of the systems is not available, then none of the systems can be updated. So while two-phase commit is an important and widely used technique, it is not always the right approach. When we balance the needs for performance, availability, and consistency, we find that there are a range of options for each. Providing an absolute consistency can decrease performance and availability. There are many excellent books devoted to transaction processing that explore this topic further—see [3] as an example.

Convergent consistency is an alternative way to think about providing consistency across systems. The basic idea of convergent consistency is that if we have a distributed set of systems that we want to keep synchronized, whenever we apply an update to one system, that update gets forwarded to all of the other systems. There are a variety of ways to do this—we could do this as each change occurs (which can result in a lot of communications traffic), or we could accumulate a set of changes and process them a batch at a time. Passing along the changes as they happen allows the receiving systems to be only a few updates behind the system that was directly updated—but it can be costly in terms of resources. Processing a batch at a time means that the changes will be delayed in getting to the other systems, but fewer resources will be consumed. With either approach, if new updates stop arriving, all of the systems will eventually have the same data. That is, the information in the different systems converge, and all of the systems become consistent with one another. The benefit of following a convergent consistency strategy is that the systems can operate independently of each other so that processing of forwarded updates can happen at the convenience of the recipient. This fact means that we can achieve higher availability and potentially higher performance at the expense of consistency. The lag-time for changes to propagate across all of the systems can be tuned by increasing or decreasing the rate at which the changes are forwarded. One other consideration is that if new updates are being applied to multiple systems concurrently, significant care must be taken to prevent anomalous behaviors such as update conflicts.

Absolute and convergent consistency are both important strategies for managing replicated data across multiple systems. Absolute consistency is not always technically possible or pragmatic. Many systems do not expose interfaces that support two-phase commit. Convergent consistency can be quite pragmatic and can yield better performance and availability, but it also has its share of complexity. System architects implementing MDM Systems need to be well versed in these techniques to properly select the right combinations of techniques that will balance the requirements and constraints dictated by an implementation.

1.3.5 MDM Implementation Styles

MDM Systems are implemented to improve the quality of master data and to provide consistent, managed use of this information in what is often a complex and somewhat tangled

environment. There are a variety of ways to support these requirements in ways that accommodate a range of methods of use (as described earlier) and implementation requirements. Implementation requirements can dictate:

- If the MDM System is to be used as a system of record or a system of reference
- If the system is to support operational environments, decision support environments, or both
- If it is important for the MDM System to push clean data back into existing systems
- If the system is to be part of an SOA fabric
- If geographic distribution is required

Different combinations of implementation and usage requirements have led to the evolution of a number of MDM implementation styles. Hybrid implementations that combine multiple implementation styles are common. Because some styles are simpler than others, organizations may start with a simpler implementation style that addresses the most urgent business needs and then subsequently address additional business needs by extending the implementation to enable additional styles.

In this section, we introduce four common implementation styles:[11]

- Consolidation Implementation Style
- Registry Implementation Style
- Coexistence Implementation Style
- Transactional Hub Implementation Style

As the styles progress from Consolidation Implementation Style to Transactional Hub Implementation Style, they provide increasing functionality and also tend to require more sophisticated deployments.

1.3.5.1 Consolidation Implementation Style

The **consolidation** implementation style brings together master data from a variety of existing systems, both databases and application systems, into a single managed MDM hub. Along the way, the data is transformed, cleansed, matched, and integrated in order to provide a complete **golden record** for one or more master data domains. This golden record serves as a trusted source to downstream systems for reporting and analytics, or as a system of reference to other operational applications. Changes to the data primarily come in from the systems that feed it; this is a read-only system. Figure 1.12 illustrates the basic consolidation style, with reads and writes going directly against the existing systems and the MDM System (in the middle) receiving updates from these existing systems. The integrated and cleansed information is then distributed to downstream systems (such as data warehouses) that use, but don't update, the master data.

There is a strong similarity between the consolidation implementation style and an operational data store (ODS). An ODS is also an aggregation point and staging area for analytical systems such as data warehouses—see [1,4] for more details. The distinction between them

11. A detailed discussion on the capabilities of these styles may be found in Chapter 3. Chapter 5 describes the implementation patterns associated with these implementation styles.

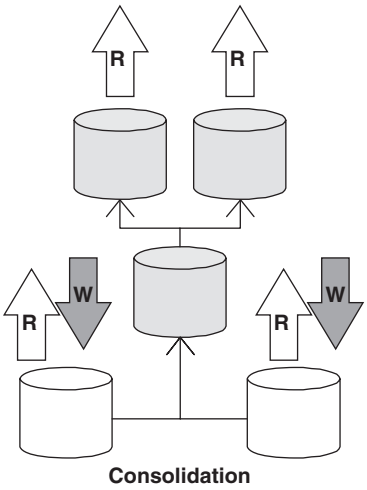

Figure 1.12 Consolidation Implementation Style.

lies in the set of platform capabilities that an MDM System offers, which go beyond the storage and management of data that an ODS provides. An operational data store is a database that is used in a particular way for a particular purpose, while an MDM System provides access, governance, and stewardship services to retrieve and manage the master data and to support data stewards as they investigate and resolve potential data quality issues.

Implementing the consolidation style is a natural early phase in the multiphase roll-out of an MDM System. A consolidation style MDM system serves as a valuable resource for analytical applications and at the same time provides a foundation for the coexistence and transactional hub implementation styles.

The drawbacks of the consolidation style mirror its advantages. Because it is fed by upstream systems, it does not always contain the most current information. If batch imports are performed only once a day, then the currency requirements for a decision support system would likely be met—but those for a downstream operational system may not be. Because the consolidation style represents a read-only system, all of the information about a master data object must already be present in the systems that feed the MDM System. Thus, if additional information needs to be collected to address new business needs, one or more of the existing source applications need to be changed as well as the MDM System—this lack of flexibility is addressed by the coexistence and transactional hub implementation styles.

1.3.5.2 Registry Implementation Style

The **registry** implementation style (as shown in Figure 1.13) can be useful for providing a read-only source of master data as a reference to downstream systems with a minimum of data redundancy. In the figure, the two outside systems are existing sources of master data. The MDM System in the middle holds the minimum amount of information required to

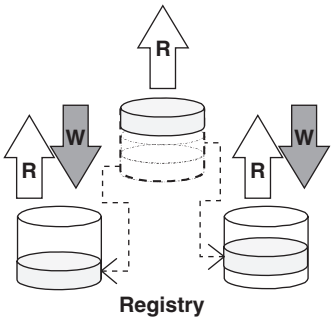

Figure 1.13 Registry Implementation Style.

uniquely identify a master data record; it also provides cross-references to detailed information that is managed within other systems and databases. The registry is able to clean and match just this identifying information and assumes that the source systems are able to adequately manage the quality of their own data. A registry style of MDM implementation serves as a read-only system of reference to other applications.

Queries against the registry style MDM System dynamically assemble the required information in two steps. First, the identifying information is looked up within the MDM System. Then, using that identity and the cross-reference information, relevant pieces of information are retrieved from other source systems. Figure 1.14 shows a simple example where the MDM System holds enough master data to uniquely identify a customer (in this case, the Name, TaxID, and Primary address information) and then provides cross-references to additional customer information stored in System A and System B. When a service request for

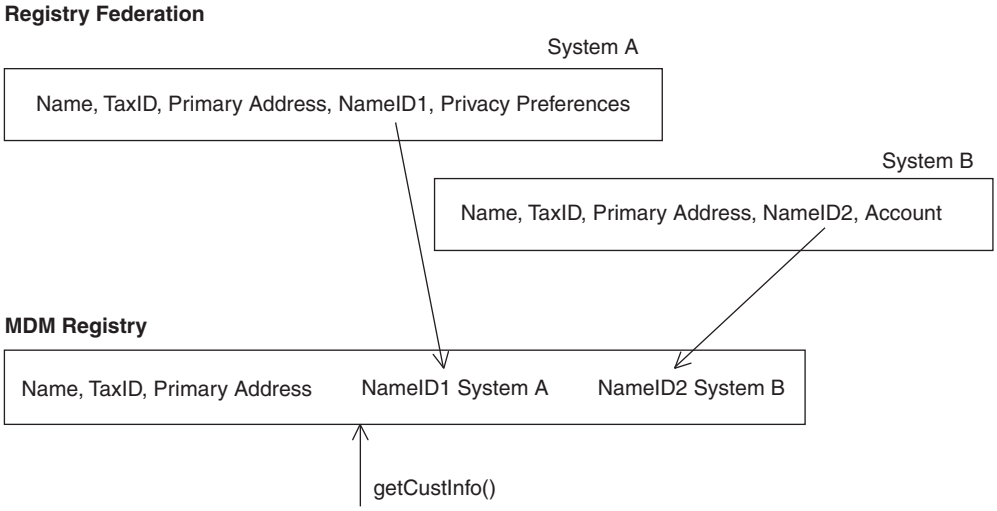

Figure 1.14 MDM Registry Federation.

customer information is received (*getCustInfo()*), the MDM System looks up the information that it keeps locally, as well as the cross-references, to return the additional information from Systems A and B. The MDM System brings together the information desired as it is needed—through federation. Federation can be done at the database layer or by dynamically invoking services to retrieve the needed data in each of the source systems.

Federation has several advantages. Because the majority of information remains in the source systems and is fetched when needed, the information returned is always current. This style of MDM System is therefore suitable to meet transactional inquiry needs in an operational environment. The registry implementation style can also be useful in complex organizational environments where one group may not be able to provide all of its data to another. The registry style can be relatively quick to implement, because responsibility for most of the data remains within the source systems.

There is, however, a corresponding set of issues with this implementation style. One fundamental issue is that a registry implementation is not useful in remediating quality issues that go beyond basic identity. A registry implementation can only manage the quality of the data that it holds—so while it can match and cleanse the core identifying data, it cannot, in itself, provide a completely standardized and cleansed view of the master data. Because the complexities of updating federated information lead most registry style implementations to be read-only, the cleansed identifying information is not typically sent back to the source systems. If the data in the sources systems is clean, the composite view served by the MDM System will also be clean. Thus, a registry implementation can act as an authoritative source of master data for the key identifying information that it maintains.

A registry implementation style is also more sensitive to the availability and performance of the existing systems. If one of the source systems slows down or fails, the MDM System will be directly affected. Similarly, the registry style also requires strong governance practices between the MDM System and the source systems, because a unilateral change in a source system could immediately cause problems for users of the MDM System. For example, in the scenario shown in Figure 1.14, suppose a change is made in the structure of the Privacy Preferences information in System A. If this change occurs without making corresponding changes in the MDM System, then a request such as *getCustInfo()* will likely cause the MDM system to fail with an internal error because of the assumptions it makes about the structure of the data it federates.

1.3.5.3 Coexistence Implementation Style

The coexistence style of MDM implementation involves master data that may be authored and stored in numerous locations and that includes a physically instantiated golden record in the MDM System that is synchronized with source systems. The golden record is constructed in the same manner as the consolidation style, typically through batch imports, and can be both queried and updated within the MDM System. Updates to the master data can be fed back to source systems as well as published to downstream systems. In a coexistence style, the MDM System can interact with other applications or users, as shown in Figure 1.15.

An MDM System implemented in the coexistence style is not a system of record, because it is not the single place where master data is authored and updated. It is a key participant in

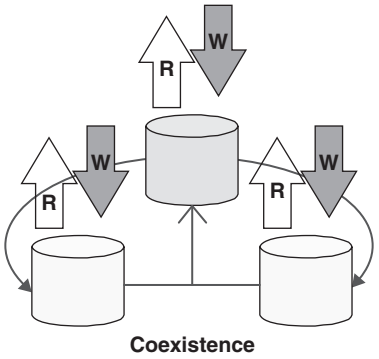

Figure 1.15 Coexistence Implementation Style.

a loosely distributed environment that can serve as an authoritative source of master data to other applications and systems. Because the master data is physically instantiated within the system, the quality of the data can be managed as the data is imported into the system. If the MDM System does a bidirectional synchronization with source systems, care must be taken to avoid update cycles where changes from one system conflict with changes from another—these cycles can be through a combination of automated and manual conflict detection and resolution.

The advantage of the coexistence style is that it can provide a full set of MDM capabilities without causing significant change in the existing environment. The disadvantage is that because it is not the only place where master data may be authored or changed, it is not always up to date. As with the consolidation style, the coexistence style is an excellent system of reference but is not a system of record.

1.3.5.4 Transactional Hub Implementation Style

A **transactional hub** implementation style is a centralized, complete set of master data for one or more domains (see Figure 1.16). It is a system of record, serving as the single version of truth

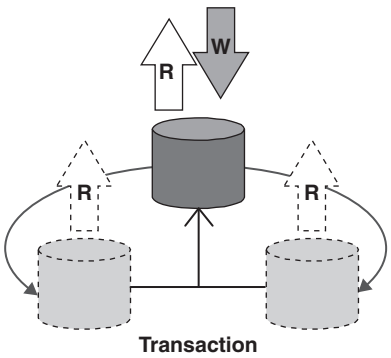

Figure 1.16 Transactional Hub Implementation Style.

for the master data it manages. A transactional hub is part of the operational fabric of an IT environment, receiving and responding to requests in a timely manner. This style often evolves from the consolidation and coexistence implementations. The fundamental difference is the change from a system of reference to a system of record. As a system of record, updates to master data happen directly to this system using the services provided by the hub. As update transactions take place, the master data is cleansed, matched, and augmented in order to maintain the quality of the master data. After updates are accepted, the system distributes these changes to interested applications and users. Changes can be distributed as they happen via messaging, or the changes can be aggregated and distributed as a batch.

Sometimes data extensions are needed in the MDM System to accommodate information that is not already stored in the source systems. For example, in the product domain, a food retailer might find that consumers are interested in knowing the distance that food has traveled to a store (there is a growing interest in purchasing locally grown products). Rather than augment all of the source systems, the MDM System would be extended to support this new information and would become the only place where such information is managed.

Governance and security are key aspects of all MDM implementation styles. Access to the master data must be tightly controlled and audited. Auditing can be used to track both queries and changes to the data. Visibility of the information may be controlled to the attribute value level to ensure that the right people and applications are restricted to seeing the right information in the right context. Because a transactional hub implementation is a system of record, security and governance play an especially critical role in maintaining the integrity of the master data.

The benefits of a transactional hub implementation are significant. As the system of record, it is the repository of current, clean, authoritative master data providing both access and governance. Any of the methods of use can be implemented (collaborative, operational, and analytical) to meet the MDM needs of an organization. The primary difficulty in a transactional hub implementation is achieving the transition from system of reference to system of record. As a system of record, all updates should be funneled to the MDM System—this means that existing applications, business processes, and perhaps organizational structures may need to be altered to use the MDM System. Although potentially costly, the overall organization generally benefits as more comprehensive data governance policies are established to manage the master data.

The primary disadvantages of the transactional hub style are cost and complexity. The implementation of a transactional hub often means that existing systems and business processes have to be altered when the transactional hub becomes the single point of update within the environment. The transition to a transactional hub can be performed incrementally to minimize disruption. The significant benefits of a transactional hub implementation cause it to be the ultimate goal of many MDM projects.

The different implementation styles introduced in this section are complementary and additive. Table 1.2 provides an overview that compares the implementation styles and shows the individual benefits and drawbacks. Different MDM domains may be implemented with different styles within the same MDM System. As we have mentioned, it is common for an MDM deployment to start with one style, such as the consolidation style, achieve success with that

Table 1.2 MDM Implementation Styles

Style	Consolidation	Registry	Coexistence	Transactional Hub
What	Aggregate master data into a common repository for reporting and reference	Maintain thin system of record with links to more complete data spread across systems; useful for real-time reference	Manage single view of master data, synchronizing changes with other systems	Manage single view of master data, providing access via services
Benefits	Good for preparing data to feed downstream systems	Complete view is assembled as needed; fast to build	Assumes existing systems unchanged, yet provides read-write management	Support new and existing transactional applications; the system of record
Drawbacks	Read-only; not always current with operational systems	Read-mostly; may be more complex to manage	Not always consistent with other systems	May require changes to existing systems to exploit
Methods of use	Analytical	Operational	Collaborative, Operational, Analytical	Collaborative, Operational, Analytical
System of	Reference	Reference	Reference	Record

implementation by publishing authoritative master data to downstream systems, and then extend the system with a coexistence style. With the completion of the coexistence phase, the MDM System could then be used to support the master data needs of new applications while continuing to publish snapshots of master data to downstream systems. Over time, the existing systems could be altered to leverage the MDM System, which would become a system of record.

In an MDM System supporting multiple domains of master data such as customer, product, and supplier, we may find that the MDM System may appear as a consolidation style for one domain, a registry style for the second domain, and a transactional hub for the third domain.

1.3.6 Categorizing Data

There are many ways to characterize the different ways to store, manage, and use data. Because these characterizations can sometimes be confusing, we put forward a set of working

definitions for five key categories of data that we discuss in this book. The five key kinds of data that we discuss in this section are:

- Metadata
- Reference Data
- Master Data
- Transaction Data
- Historical Data

Each of these categories of data has important roles to play within an enterprise's information architecture.

1.3.6.1 Metadata

The distinctions between metadata, master data, and reference data can be particularly confusing. In this book, we use the term **metadata** to refer to descriptive information that is useful for people or systems who seek to understand something. Metadata is a very broad topic—there are thousands of different kinds of metadata. It is beyond the scope of this book to provide an in-depth review of the topic; however, we can describe a few key characteristics.[12] Different kinds of metadata are defined and used pervasively throughout the software industry because it is useful to be able to have one kind of information describe another kind of information. A database catalog describes the data managed within a database, an XML schema[13] describes how an XML document that conforms to the schema should be structured, and a WSDL[14] file describes how a Web service is defined. Metadata is used in both runtimes and in tools. For example, a relational database uses metadata (the database catalog) to define the legal data types for a column of data, that is, to recognize if a column of information is a primary key and to indicate if values in a column of data can be null. Similarly, database tooling uses this same metadata to allow database administrators to author and manage these database structures.

In general, it is considered appropriate to hide the existence of metadata by making the creation, management, and use of metadata part of the systems and tools that need to use it. For example, many different tools and runtimes are involved in collecting data quality information. This metadata helps users determine how much they should trust the data and systems monitored. It is important that the collection and processing of this information be automated and transparent to users. If it is something that requires user involvement, then it is difficult to guarantee that the collection of the quality metadata has been done in an accurate and consistent manner. Similarly, most users should not have to explicitly recognize when they are using metadata—it should just be a natural part of their work environment.

Metadata is also stored and managed in a wide variety of ways—from files to specialized metadata repositories. Metadata repositories often provide additional benefits by allowing different kinds of metadata to be linked together to promote better understanding and to

12. While the topic of metadata management is very broad, one perspective can be found in [5].

13. XML Schema (XSD) is a standard that describes how XML may be structured. See [6] for a technical overview of XSD.

14. Web Service Description Language is a standard way to describe Web services. Chapter 2 will describe WSDL in more detail.

support impact analysis and data lineage across a range of different systems. For example, because there are many places where data quality information can be collected and exploited, it is useful to aggregate this information into a Metadata Repository where the information can be combined, related, and accessed by multiple tools and systems. An increasingly important new kind of metadata repository is a **Service Registry and Repository (SRR)**, which specializes in storing information about the services deployed in an SOA environment to support both the operation and management of a services infrastructure.

Because metadata is data that describes other kinds of information, there is metadata for each of the other kinds of data. For example, in the case of master data, the information model and services provided by an MDM System are described by a set of metadata that is used at both design and execution. Users of the MDM System rely on this metadata to accurately describe how to interpret and use the master data. When metadata plays such a critical role, governance of the metadata is important to users' confidence that the metadata accurately reflects the MDM System.[15]

1.3.6.2 Reference Data

Where metadata often describes the structure, origin, and meaning of things, **reference data** is focused on defining and distributing collections of common values. Reference data enables accurate and efficient processing of operational and analytical activities by enabling processes to use the same defined values of information for common abbreviations, for codes, and for validation. Reference data can be simple lists of common values to be used in lookups to ensure the consistent use of a code such as the abbreviation for a state, of product codes that uniquely identify a product, or of transaction codes that specify if a checking transaction is a deposit, withdrawal, or transfer. In all of these cases, the reference data represents an agreed-upon set of values that should be used throughout an organization.

Reference data is used throughout an IT system—including the processing of financial transactions, the analysis of data in a warehouse, and the management of the systems themselves. Wherever we want to guarantee a common value of a simple object, we are using reference data. When the values of reference data are able to change during the processing of long-running business processes, the management of reference data becomes particularly important. For example, when two companies merge, the stock symbol representing them may change, and so all in-flight transactions that referenced that stock symbol might fail unless the reference data is managed appropriately.

It is often the case that different applications have different values for the same object. For example, one application may refer to a state by its abbreviation, and another application may refer to it by its full name (e.g., TX and Texas). A reference data management system helps to translate from one set of values to another.

The management of reference data can happen in multiple places. Many applications have their own built-in management for reference data. Dedicated reference data management systems are sometimes used for specialized forms of reference data, and in particular for the

15. Chapter 9 describes governance in detail.

reference data used in financial investment transactions. Some MDM Systems can also be used to manage reference data in addition to master data.

1.3.6.3 Master Data

As we described earlier, master data represents the common business objects that need to be agreed on and shared throughout an enterprise. In previous sections, we described the domains of master data and the methods of use. Master data is most often managed within a specialized MDM System. An MDM System often uses both reference data and metadata in its processing. Reference data is used to ensure common and consistent values for attributes of master data such as a country name or a color. MDM Systems may either store metadata internally or leverage an external metadata repository to describe the structure of the information managed by an MDM System and the services it provides.

1.3.6.4 Transaction Data

The business transactions that run an organization produce transaction data. **Transaction data** is the fine-grained information that represents the details of any enterprise—in the commercial world this information includes sales transactions, inventory information, and invoices and bills. In noncommercial organizations, transaction data might represent passport applications, logistics, or court cases. Transaction data describes what is happening in an organization.

There is often a relationship between transaction data and master data. For example, if a person applies for a passport, the application and the processing of the application is transaction information that refers to the master data representing people. The master data contains citizenship status, existing passport details, and address information that is needed in the processing of the passport request. The processing of the passport itself is handled by other applications.

Transaction data is usually maintained in databases associated with the applications that drive the business and that may also be geographically and organizationally distributed. An organization can have a very large number of databases with transaction databases—each holding a large amount of data. Transaction data is commonly stored in relational databases according to schemas that have been optimized for the combination of query and update patterns required.

1.3.6.5 Historical Data

Historical data represents the accumulation of transaction and master data over time. It is used for both analytical processing and for regulatory compliance and auditing. Data integration tooling is typically used to extract transaction data from the existing application systems and load it into an ODS.[16] Along the way, it is often transformed to reorganize the data by subject, making it easier for the data to be subsequently loaded into a data warehouse for reporting and analysis. The ODS may be updated periodically or continuously.

The historical data loaded into the warehouse is used to gain insight into the functioning of the business. Many different kinds of analyses may be performed that can support a wide range of

16. In some environments, transaction data may be directly loaded into a data warehouse or mart instead of, or along with, loading into an ODS. Please see [4] for more discussion.

Table 1.3 Key Data Characteristics

	What Kind of Information?	Examples	How Is It Used?	How Is It Managed?
Metadata	Descriptive information	XML schemas, database catalogs, WSDL descriptions Data lineage information Impact analysis Data Quality	Wide variety of uses in tooling and runtimes	Metadata repositories, by tools, within runtimes
Reference Data	Commonly used values	State codes, country codes, accounting codes	Consistent domain of values for common objects	Multiple strategies
Master Data	Key business objects used across an organization	Customer data Product definitions	Collaborative, Operational, and Analytical usages	Master Data Management System
Transactional Data	Detailed information about individual business transactions	Sales receipts, invoices, inventory data	Operational transactions in applications such as ERP or Point of Sales	Managed by application systems
Historical Data	Historical information about both business transactions and master data	Data warehouses, Data Marts, OLAP systems	Used for analysis, planning, and decision making	Managed by information integration and analytical tools

uses, including basic reporting, dashboards (which show key performance indicators for a specific part of the organization, and predictive analytics that can drive operational decisions).

Historical data is also required to conform to the wide variety of regulations and standards that organizations have to comply with. As described throughout many sections of this book,[17] the regulatory environment drives the need for the management of historical as well as master data.

Table 1.3 summarizes some of the key characteristics of these different kinds of information.

17. Including Section 1.4.2.2, Chapter 9, and Appendix C.

1.4 Business Benefits of Managed Master Data

In this section, we take a look at some benefits of using managed master data within an MDM System. We will categorize these benefits using the same three principles we discussed earlier, that is, that an MDM System:

- Provides a consistent understanding and trust of master data entities
- Provides mechanisms for consistent use of master data across the organization
- Is designed to accommodate and manage change

In Chapters 6–8, we focus on Solution Blueprints and discuss specific examples of the business benefits of MDM within particular industries and problem areas.

1.4.1 Consistent Understanding and Trust of Master Data Entities

One of the main objectives of MDM enablement of an enterprise is to improve the quality of the master data elements for the entire business. Here, we look at some different aspects of data quality and how MDM can help improve them. With higher quality in the data comes a more consistent understanding of master data entities. A broader discussion on data quality can be found in Chapter 9.

1.4.1.1 Accuracy

"Bad data costs money." High-quality data is required for sound operations, decision making, and planning. Data quality is high if the data correctly represents the real-life constructs to which it refers. A traditional development rule states that $1 spent in design costs $10 in development and $100 in support [7]. The same applies to data quality: Money and effort spent designing for higher data quality early on provides a good return on investment when compared to fixing data issues later on. Even this is a fraction of the cost of having to deal with data quality issues after they have caused business problems. In reality, the $100 measurement is probably not even close to covering the cost of bad data. It is estimated by the Data Warehousing Institute that bad data costs U.S. business over $600 billion a year [8]. The cost of bad data manifests itself in misplaced shipments, product returns, lost marketing opportunities, cost for immediate and near-term system repairs, loss of customer trust, loss of customers, and an adverse impact on sales and thus market share.

An MDM Solution needs to have the capabilities to increase the quality and thereby the trust in an enterprise's data compared to the previous state of unmanaged master data. Accuracy and completeness of master data, its consistency, its timeliness, its relevance, and its validity are the primary contributors to data quality. **Accuracy** of the data is defined as the degree of conformity that a stored piece of information has to its actual value. Because of the central positioning of master data within the enterprise, the accuracy of master data, in particular, plays a large role in determining the overall data quality. Data that can be validated while a customer service representative is on the phone with a client can be made more accurate than it was before by just updating a "validated-on-date" field. There is no change to the actual data, but it is now known to be accurate. Thus, knowing when the data

has changed can be at least as important as the change itself. The accuracy of the data is also dependent on the context. Data that is sufficiently accurate in one context may not be accurate enough for another context. For example, a value for "age" may suffice for marketing purposes, whereas a legal document may require a date of birth.

Higher accuracy of data leads to greater efficiencies in business processes that use the data. Because of its high degree of reuse throughout the enterprise, this relationship is especially true for master data. Major improvements in data accuracy can be achieved through MDM functionality such as matching/de-duplication and structures data stewardship.

Matching and de-duplication refers to taking data from multiple source systems, multiple channels, or different interactions and matching them up with existing data in the MDM System. Before we begin the matching process, we first validate and standardize the data to improve the accuracy of the matching. Matching typically involves creating a candidate list of possible matches (also referred to as **bucketing**) based on the data already in the system and then comparing these candidates against the incoming record. By calculating a score for each comparison, the matches can be ranked. Typically, a threshold value indicates that certain records indeed match, and now these records can be collapsed together into a new combined record using a set of survivorship rules for each of the attributes of that record.

Data stewardship involves human interaction to determine match and no-match cases where automated processing cannot provide a guaranteed match. This process means manually assessing the possible matching and de-duplication and subsequently deciding on the survivorship of the individual attributes. Data stewardship and the management of data quality are discussed further in Chapter 9.

Data validation is a technique to improve the accuracy and precision of the data by using a set of rules to determine whether data is acceptable for a system. Examples are data formats, range validations, limit checks, and checksums on a data element, or cross-checks on multiple data elements. In an MDM context, there may be a validation rule to describe the format of a product identifier or to validate that a person can only have, at most, one legal name.

Because the master data in an MDM System often comes from multiple source systems, and because the validation rules across those source systems are seldom fully synchronized, the MDM System provides its own superset of data validation rules. Data validation can be enforced as part of an Extract, Transfer, Load (ETL) process (more on this later) when data is loaded into the MDM System. However, data validation also needs to be part of online transaction processing to validate data when MDM services are invoked. Centralizing data validation in an MDM System achieves a higher level of data validation for the enterprise as a whole.

1.4.1.2 Completeness

Completeness of master data is determined by the degree to which it contains all of the relevant entities, attributes, and values required to represent the real-life master constructs such as customers, products, or accounts. Typical questions asked in this regard would be whether all of the entities for a given master construct are present, including all of the

required attributes for these master data entities and their values. For example, in an MDM System managing customer information, are all of the required addresses (shipping, billing, or vacation) available? Do they contain the required attributes (address line city, postal/zip code) and are the values for these attributes provided?

Completeness is also dependent on business context—what is required in one context might be optional in another. For example, in the case of a life insurance customer, smoking status is a mandatory attribute, and the master data record for this customer would not be complete without a valid value for this mandatory attribute. The same customer record in the context of car insurance can be considered complete without this attribute. An MDM System servicing both verticals would need to be flexible enough to support this contextual distinction. MDM Systems typically have many different contributors of data within an enterprise. This enables an MDM System to maintain a more complete picture of the master data than any of the contributing systems on their own, because each keeps only a subset of the total data as required for their business purpose. In other cases, the MDM System might be the system where the collective data from the source systems is augmented with additional information not kept in any of the source systems.

1.4.1.3 Consistency

Borrowing from the general notion of **consistency** in formal logic, we can define master data to be consistent when data retrieved through two different locations, channels, applications, or services cannot contradict itself. In other words, at no time should the manner by which the data is accessed have an effect on the information it represents. In a consistent environment, the values for data should be the same.

This might seem obvious, but as we saw in the previous section, the initial scattering of unmanaged master data across the enterprise is often natural and more or less unavoidable. What is also unavoidable is that the quality of that master data differs from system to system. It is therefore entirely possible that these sources disagree on a particular aspect of a master data object. For example, what one system has stored as a billing address might be kept as a shipping address in the other. Similarly, the date of birth of a customer in one source system may be different from that in another. A billing system might have accurate account and address information but would not be the trusted source for date of birth or e-mail address, while an online self-service system would have more accurate e-mail information but not necessarily the best postal addresses.

Even when we only focus on a single system, there can be large variations in data quality due to variations in the level of data consistency. All of these types of inconsistencies need to be addressed by the MDM System, both at the time of deployment and throughout its lifecycle. Being centrally positioned within the enterprise, an MDM System is in a unique position to improve the consistency of master data for an entire enterprise. Consistency is also determined by the level of standardization, normalization, and validation that was performed on the data. Data standardization ensures that the data adheres to agreed-upon guidelines. For example, address standardization determines what an address should look like for a specific geography and gives a fixed format based on a postal code look-up. Many other elements, such as first and last names, can also be standardized. Standardization

greatly improves the ability for computer systems to locate and manipulate data elements. Data normalization describes the organization of data elements in related subcomponents. This can be thought of in the traditional context of the database modeling technique of normalization but also on a much smaller scale, for example, parsing a personal name such as "MARIA LUZ RODRIGUEZ v. de LUNA" into the correct data structures.

As we saw earlier in Section 1.3.5, some MDM implementation styles are more prone than others to show some level of data inconsistency. In essence, as soon as master data appears in multiple places, there is a potential for data inconsistencies. This is true even if these sources are managed replicas. Replication technology is very good at being able to keep multiple copies synchronized—but there is often some amount of lag between copies as changes take place.

1.4.1.4 Timeliness

The timeliness of master data is another important factor determining its quality. Master data changes relatively slowly compared to other forms of business data. For example, we have observed that in financial institutions, customer data changes around three percent per day and that contract-related information can change eight percent per day. Address and phone numbers for individuals seem to change, on average, every 2.5 years. Product information in retail can change quickly as retailers introduce seasonal products into their catalogs.

These changes often take time to propagate through the enterprise and its systems. With this propagation comes a delay between the data being changed and the availability of this change to the data consumers—the longer the delay, the greater the potential loss in data quality.

A typical example of this is a traditional data warehouse where data is extracted from source systems, cleansed, de-duplicated, and transformed for use in an analytical context. Because many data warehouses are used for off-line decision support, it is common for them to be updated on a daily or sometimes weekly basis. Thus, the data is always somewhat out of date with respect to the operational systems that feed it. Such warehouses may not be suitable for operational usage. An MDM System may take on the task of maintaining this cleansed version of the master data on an ongoing basis, while serving as a source for the data warehouse. In this case, the MDM System is providing on-line access to this cleansed data.

Another factor in the timeliness of the data is its freshness. Captured data typically deteriorates by a certain rate because the real life constructs it represents change over time and not all these changes make it back into the captured data. For example, on average about 20% of all Americans change their address every year.

Timeliness thus affects both consistency and accuracy. Propagation delay impacts the consistency of the information across systems, whereas freshness is one indication of how accurate the data remains.

1.4.1.5 Relevance

Data **relevance** is a measure of the degree to which data satisfies the needs of the consumer. It measures how pertinent, connected, or applicable some information is to a given matter.

What is obvious from this definition is that relevance is also context-sensitive. Master data that is relevant in one context or to one user might be irrelevant in another context or to another user. If all relevant information is captured for the different consumers of the information, then the information can be considered to be complete.

For example, the physical dimensions of a grocery item in a product information system are relevant to someone in shipping but irrelevant to a translator who works on creating a Spanish version of the product catalog. In the description of completeness, we discussed the example of a smoker status and its relevance to different lines of business in an insurance company. Data relevance determines why and what we measure or collect. To ensure data relevance, the "noise" (unnecessary information) factor of the data needs to be reduced.

In the case of operational data, relevance is usually determined during the definition and requirements phase of an MDM project. For example, during this phase of an MDM implementation, a gap analysis can be used to determine which relevant data elements need to be added to an existing data model. If data from multiple source systems is combined in an MDM System, then the relevance of the data elements from a single system to the enterprise as a whole needs to be determined.

The MDM System does not necessarily need to contain the sum of all of the parts. Certain pieces of data might be irrelevant from an enterprise point of view. These additional data elements may continue to be maintained in the line-of-business systems. Relevance may also change over time. As business needs change, what is relevant today may change tomorrow. Accommodating these changes is a natural part of the evolution of an MDM System.

1.4.1.6 Trust

We can trust data when we know that it has met an appropriate set of standards for accuracy, cleanliness, consistency, and timeliness—that is, when we know that data stewards manage the data and that the data is protected from unauthorized or unmanaged updates. The more we know about the data, the more we understand the data itself and what is meaningful (and what is not), and the more we learn how to gauge our trust in the information. We can learn about the data and data quality using data profiling tools, and we can begin to understand the provenance or lineage of the data through a combination of automated and manual techniques.

We can aggregate master data from across the enterprise, clean it, reconcile it, and then manage it so that we control who is allowed to see it and who is allowed to change it. By actively managing master data in this way, we can assert our trust in this master data. When we believe that we can trust the data—that we manage and maintain a collection of trusted information—then we can be an authoritative source of that data for other users and applications.

1.4.2 Consistent Use of Master Data Across the Organization

It is not just the quality and consistency of the data that is important—it is also the consistent usage of that master data throughout the enterprise. MDM Systems offer a consistent,

comprehensive view of master data across the organization. Typically, this unified view is not available before an MDM implementation takes place. In this pre-MDM situation, master data is typically spread out across multiple, autonomous line-of-business systems. These systems could be of a homogeneous or heterogeneous nature. An example of a homogeneous situation will be presented in Chapter 8, where we will see a solution blueprint using an MDM System to provide a consistent view of product data across SAP systems from multiple geographies. A heterogeneous example could be one where customer data is stored both in Siebel CRM and in a custom-built billing system. The benefits of using MDM in this situation come from the data quality improvements we saw earlier and from cost savings and efficiencies we will describe later, as well as from improved support for regulatory compliance.

1.4.2.1 Cost Savings and Efficiencies

Cost reduction and avoidance is another benefit of tackling MDM. There are many operational savings and efficiencies that can be achieved by implementing reusable services supporting key processes such as name and address change in CDI or product information changes in PIM. In an unmanaged master data environment, transactions like these typically need to be applied to every application that contains such data, often by manually re-keying information. Depending on the MDM implementation style in use, such processing can be drastically reduced by only updating the coexistence or operational MDM hub and automatically forwarding such changes to the interested applications, or by having these other applications consult the MDM System directly for this master data. This process can reduce the amount of effort required to propagate such changes through the enterprise and improves this propagation by ensuring all relevant systems are updated. This process also improves the quality of the propagation by ensuring that all the updates are the same and that no re-keying errors occur.

Other cost avoidance opportunities can be identified by focusing on the benefits of having an enterprise-wide view of the master data entities available in the MDM Solution. This enterprise view allows for the discovery of relationships between entities that were previously only distributed across multiple systems. In the case of CDI, we can therefore bring together all of the information about a party from across all of the different systems, including all of the addresses at which the client can be reached, all of the products the client owns (obtained from different lines of business), family relationships, or additional identifiers such as driving license or passport number. All of this information can be used to optimize dealings with the client and provide a better customer experience in dealing with the company—thereby increasing customer retention.

MDM can lead to a reduction in data storage costs and total cost of ownership of a solution by removing redundant copies of master data, although this benefit mainly occurs in a consolidation and transactional hub style of MDM. The data volumes occupied by master data in a typical enterprise are very significant and there can be substantial savings in storage costs. In a registry or coexistence style MDM implementation, the storage requirements typically increase, because all of the existing unmanaged master data copies are still maintained, together with the new data storage requirements for the MDM hub. Another caveat

here is that in many cases an MDM System starts to store more master data than was originally available in unmanaged form. In essence, the master data in the MDM System is augmented with data previously not recorded in the enterprise. Data fields may need to be larger—for example, the ID field needs to be longer because more entities appear in a single system. Typical examples can be found in the area of privacy preferences or e-mail addresses that were previously not recorded in the older source systems.

Enterprise resources such as money, labor, advertising, and IT systems are typically scarce commodities that need to be applied in those areas where they can offer maximal return on investment. Unmanaged master data hinders this resource allocation because the information required to drive decisions is scattered among many systems. Questions such as "Who are my most valuable customers?," "Which are my best selling products?," and "Is there fraud—and if so, where?" require a managed set of master data.

In traditional operating environments, such decisions are based on information from a data warehouse, which often is the only location where data was available in cleansed, unduplicated form. Data warehouses, however, have inherent design characteristics to optimize them for analytics and reporting, and they are generally not designed to support operational transactions. In addition, this data often has a higher degree of latency and is therefore somewhat stale. Without MDM, a customer who just bought a high-end product over the Web and is calling in to the call center might not appear as a very valuable client to the customer service representative. Without managed master data, it is very difficult to get a complete view of such a customer or product and determine its value to the enterprise. Consequently, resources can't be optimally applied, and it is difficult to provide a higher level of service.

Supporting all channels with managed master data delivers common, consistent information that allows customer service representatives to give the same discount to a client on the phone as the one he or she just handled by mail or on the Web site. Managed master data allows for the product description in the printed catalog to match the one on the Web site and printed on the product. The consistency MDM offers leads to cost savings because it reduces the effort required to process this data at a channel level. This reduction in effort is a significant improvement over the effort required to keep consistency across channels in an unmanaged master data environment.

The overall picture that MDM creates of the master data is more complete than any of the pictures in the contributing systems, and it is therefore more useful for due diligence processes or to detect potential fraud. In fact, we can use MDM to proactively uncover fraud and to create alerts or take appropriate actions. After the master data is managed by an MDM system, we can determine relationships between master data entities that were not detectable before. For example, it is very valuable to detect that a new prospective client is co-located at an address of another customer with a very similar name who is on the bankruptcy list, or to figure out that the manager in charge of purchasing is married to one of your biggest vendors.

MDM allows for the streamlining and automation of business processes for greater efficiency. Furthermore, MDM centralizes the master data within the enterprise and enables the

refactoring and reuse of key business processes around that master data. For example, CDI facilitates the development of enterprise-wide New Account Opening processes, and PIM enables the development of enterprise-wide New Product Introduction processes.

1.4.2.2 Regulatory Compliance

Many newspaper articles commence with "Since Enron and September 11, 2001" when discussing regulatory compliance, but regulatory bodies have been around for much longer than this. The original Anti-Money Laundering (AML) controls were implemented in the Bank Secrecy Act of 1970 [9] and have been amended up to the present. The Basel Committee first came together in 1975 as a result of the failure of Bankhaus Herstatt [10]. Since the two mentioned incidents, however, the pace and rigor of new regulations has increased significantly. In addition, it is very rare for one of these regulations to be withdrawn and disappear. Since 1981, over 100,000 regulations have been added in the United States [11]. Consequently, the number of regulations that a modern enterprise needs to adhere to is continually increasing, as are the expenses associated with compliance. According to a study by the Financial Executives Institute, companies should expect to spend an average of $3 million to comply with section 404 of the Sarbanes-Oxley Act (SOX) [12]. Forrester Research estimates the five-year cost of a Basel II implementation for the largest banks to be $150 million. Obviously, there are vast differences between all these regulations: by industry and by geography, and also by the strictness, penalties, and consequences of noncompliance. Some of the most well-known regulations are Basel II, Sarbanes-Oxley (SOX), the Patriot Act, Office of Foreign Assets Control (OFAC) watch lists,[18] Solvency II, "Do not call" compliance from the Federal Communications Commission (FCC), Anti-Money Laundering (AML), and HIPAA.[19]

Some of these regulations are global, while others are specific to North America or Europe, but most have equivalent regulations across all geographies. Obviously, implementing point solutions for each of these regulations separately is not viable, and enterprises have to look at approaching this situation in a more holistic manner. Fortunately, even though they have different policies, many regulations share common objectives that require that authoritative data is used in business processes and that proper controls over key data are in place. Organizations are therefore starting to establish regulatory frameworks instead of addressing each of these regulatory compliance initiatives with a dedicated point solution. This approach also better positions them to adapt to changes in regulation or the introduction of new regulations in the future. Achieving regulatory compliance initially does not add anything to a company's bottom line; it is a pure cost and used as a "penalty avoidance" measure. However, the solutions that are put in place to achieve compliance can be leveraged to drive many other advantages and differentiators in the market. Thus, as you can see in Figure 1.17, the more maturely compliance is handled, the more business value for the stakeholders can be derived from it.

18. Threat and fraud scenarios are discussed further in Chapters 4 and 7.
19. Health Insurance Portability and Accountability Act.

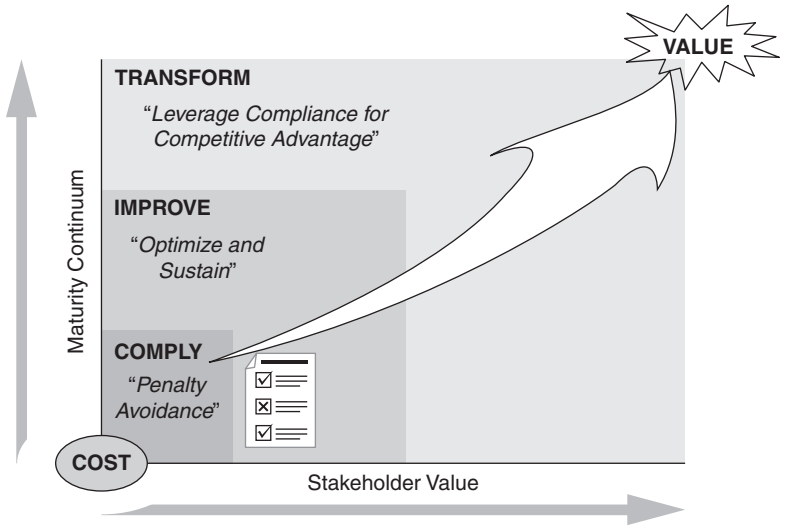

Figure 1.17 Maturation from risk mitigation and penalty avoidance to leveraging risk and compliance as a competitive advantage.

To illustrate the relationship between compliance and master data more concretely, we now describe a few of the high-impact regulatory policies.

- **CDI—Know Your Customer (KYC)**
  KYC is a compliance policy related to the Bank Secrecy Act and the USA Patriot Act and to international standards such as Solvency II and the International Accounting Standards. It requires financial institutions to diligently identify their clients and obtain certain relevant information required to enact financial business with them. One aspect of KYC is to verify that the customer is not on lists of known fraudsters, terrorists, or money launderers, such as the Office of Foreign Assets Control's (OFAC) Specially Designated Nationals list. Another is to obtain an investment profile from customers to identify their risk tolerance before selling them investment products. CDI systems are designed to store and maintain identifying pieces of information, such as driving licenses, passports, and Social Security Numbers on the parties in the system. Through the de-duplication functionality available in an MDM System, companies have a much better chance of correctly identifying two parties as being one and the same. CDI systems can store KYC party profile information like the questionnaire answers obtained for a financial profile of the client, and CDI systems can more easily check a company's entire list of customers, vendors, employees, and so on, against any of the known felon lists.
- **CDI—Privacy**
  Privacy is defined as a basic human right in the "Universal Declaration of Human Rights," and although data privacy legislation and regulation is not very strict in the United States, it is much more rigidly defined and enforced in Canada and especially in Europe. The European Commission's "Directive on the Protection of Personal

Data" states that anyone processing personal data must comply with the eight enforceable principles of good practice. These principles state that data must be:

- Fairly and lawfully processed
- Processed for limited purposes
- Adequate, relevant, and not excessive
- Accurate
- Not kept longer than necessary
- Processed in accordance with the data subject's rights
- Secure
- Not transferred to countries without adequate protection

Verification of these principles within an enterprise requires strict management and governance of the company's master data. This governance includes both the referential and persisted storage of the data as well as management of the processes handling this data. These and other requirements originating for data privacy legislation and regulations can be serviced by using MDM System features. Access to the data needs to be restricted to those who have the rights to administer it through user authorization and authentication, and data entitlements. Private data can not be kept indefinitely, and archiving and data deletion features need to be present. Preferences for do not mail, do not call, and do not e-mail need to be available within the MDM data models in order to comply, for example, with the "National Do Not Call Registry" in the United States. These kinds of data augmentations are often easier to implement in a centralized MDM System than in silo-based administrative systems.

- **CDI/Account, Credit Risk Mitigation**
  Unfortunately, not everybody pays their bills. Credit risk is the risk of loss due to non-payment of a loan or other line of credit. Offering more credit to a particular client increases the credit risk for the company. Risk is offset against the potential gains that can be made on the loan. However, in many cases, companies cannot even clearly determine the credit risk of a particular customer they are exposed to because they do not have sufficiently cleansed and de-duplicated customer data. The same customer might exist in the system multiple times and carry credit on every instance, thereby increasing the creditor's risk. Using an MDM System to keep customer data clean and de-duplicated can help lower risk exposure for the company by providing a clear picture of the risk exposure of a particular client. Another common occurrence is that some of a company's vendors are also its clients. Realizing that these two parties are one and the same can increase a company's negotiation position when credit is drawn on one side of this relationship and offered on the other.
- **PIM and Regulatory Compliance**
  Product information is also a heavily regulated asset. Packaging information, export and customs information, ingredients lists, warning labels, safety warnings, manuals, and many other types of product information all have to adhere to format, content, and language rules that are very industry- and geography-specific. Centrally storing this data in a PIM system helps the compliance process, but it is

the collaborative workflow processes around maintaining this data in a PIM system that enable proper control of product master data to ensure regulatory compliance.

- **PIM/RFID—Regulatory Compliance—Traceability in the Pharmaceutical Industry**
  The Prescription Drug Marketing Act (PDMA) of 1988 in the United States mandates that drug wholesalers that are not manufacturers or authorized distributors of a drug must provide a pedigree for every prescription drug they distribute. This regulation was created to prevent drug counterfeiters from entering illegal and potentially dangerous products into U.S. commerce. While the implementation of this regulation is still being contested in court and was postponed in 2006, several states have stepped up their individual pedigree legislation. Most noticeably, California has adopted a requirement that the drug's pedigree be available in electronic form. Electronic Product Codes (EPC) and Radio Frequency Identification (RFID) are two promising technologies that are being used in this area. By building applications around these technologies, individual shipment lots can be tracked to ensure their pedigree. Hooking this transactional data up to the product information stored in a PIM system allows for a full 360-degree view of a product, its detailed information, and its passage through the supply chain.[20]
- **Account Domain and Regulatory Compliance**
  Today, all financial institutions globally are required to monitor, investigate, and report transactions of a suspicious nature to their central banks. They must perform due diligence in establishing the customer's identity and the source and destination of the funds. The account domain of an MDM can be used to provide references to accounts that exist elsewhere, in other back-end administrative account systems, and how they are related to parties in the system. This information can be very useful in identifying possible cases of money laundering. Alternatively, the MDM System can be used as the system of record for account information, in which case the account exists and is managed solely within the MDM System. In this second case, the transactions that are being performed on such an account need to be monitored for possible fraudulent behavior.

1.4.3 Accommodate and Manage Change

In this section, we take a look at various aspects of managing and accommodating change within an organization, viewed from an MDM perspective.

1.4.3.1 Reducing Time to Market

Marketplaces are increasingly more volatile, competitive, and risky. For businesses to participate in these markets, they must be able to respond rapidly to directional, structural, and relationship changes within their chosen industries and market sectors. Reducing the time to market for their New Product Introduction is a critical objective in this pursuit. Time to market is defined as the amount of time it takes to bring a product from conception to a point where it is available for sale. Across industries, different phases in the product development

20. A detailed discussion on RFID Track and Trace can be found in Chapter 6.

process are identified as the start or end-point of the Time to Market process. In some industries, the start is defined as the moment a concept is approved; in others, it is when the product development process is actually staffed. The definition of the end of the Time to Market measurement is also open to interpretation. In some industries, it may be defined as the handover from product engineering to manufacturing, or in other industries, it may be the moment the product is in the client's hands. Regardless of the scope of the process, what is important to a business is the relevant measurement of its Time to Market against that of its direct competitors. Getting to the market first is important for various reasons. It allows an enterprise more freedom in setting the product price, because no competitive products are available until the competition catches up. It may also allow an enterprise to obtain an early foothold and capture an initial market share before its competitors, allowing the organization to profile its brand as the industry leader in that area.

It is critical for a successful and optimal execution of the New Product Introduction that consistent, high-quality information about the product is available to all parties involved in the NPI process. Unmanaged, scattered master data about products leads to inconsistencies, inaccuracies, and therefore to delays in Time to Market for the product, providing opportunities to competitors to react and get to market first. MDM is a key enabler to the management of these collaborative workflows. By obtaining a consistent, cleansed, and accurate version of the product data in a PIM system, many NPI processes can be improved. Steps in the NPI product development process typically include checking, review, approval, and control of product structures. It is therefore critical to manage the related product information in the same manner and to provide a consistent implementation of the NPI process in the PIM MDM System.

1.4.3.2 Revenue Enhancement and Other New Opportunities

MDM provides a higher level of insight into master data, and this can be used to identify opportunities for revenue improvement. Increased insight into high-value customers through profiles, or account and interactions information, can be used to identify candidates for up-sell or cross-sell opportunities. Increased insight into master data around products can then be used to identify which up-sell and cross-sell opportunities[21] exist when selling a particular product to that customer and which bundling opportunities can be leveraged.

Events relating to master data can be analyzed to identify revenue opportunities. For example, residence changes and other life events can alert sales to potentially changing customer needs. Without MDM, there is no enterprise-wide ability to recognize and communicate such events and thus no sales actions are taken, no e-mail campaigns are directed based on such events, and no outbound telephone calls or Web offers are made. All of these result in missed revenue opportunities.

1.4.3.3 Ability to Rapidly Innovate

Companies cannot grow through cost reduction and reengineering alone. Innovation is the key element in providing aggressive top-line growth and increasing bottom-line results (see [13] for details). Innovation is the successful implementation of creative ideas within an organization.

21. A more detailed discussion on leveraging master data for cross-sell and up-sell can be found in Chapter 7.

Innovation begins with a creative idea by an individual or a team, and while the initial idea is a necessary input, it is not sufficient to guarantee innovation (see [14] for details). To achieve innovation, the implementation of the idea needs to be successful. It is in the implementation of those creative ideas where MDM can help an organization innovate. Innovation within an enterprise can take many different forms. Product, service, and process innovation are some of the more obvious types, but marketing innovation, business model innovation, organizational innovation, supply chain innovation, and financial innovations are other examples of innovations that can contribute to increased success. All of these innovations have dependencies on the master data available within the enterprise and the processes surrounding them.

1.4.3.4 Product or Service Innovation

If a company wants to introduce an innovation around a product or a service it offers, or if it wants to start offering a new product or service, an MDM System can help centrally manage the related changes that need to be made to the product master data. Where no MDM System exists, product data may be scattered across the enterprise. Integrating across these multiple copies and ensuring all copies are properly updated acts as an inhibitor to innovation.

A product innovation may require updates to the MDM System as well. For example, what was previously a valid value for a product attribute might now not be incorrect, requiring a change to data validation routines. Alternatively, the innovation might require additional attributes to be kept as part of the product information, requiring changes to the data structures and metadata information. In many ways, the advantages MDM can provide here are similar to the ones that offered streamlining of the new product introduction process, as we have seen earlier. Another usage of MDM for product innovation is product bundling. Many organizations have separate lines of business that manage their individual product lines, often backed by isolated IT systems geared specifically to supporting one particular type of product. A common example here would be a telecommunications company that sells landline and mobile subscriptions, cable or satellite TV, and high-speed Internet access. Such a company might want to provide product innovation by offering its clients bundled products with associated discounts. None of the existing administrative systems may be suited for this purpose—however, an MDM System managing a combination of customer, product, and account information would be a logical starting point to enable such a purpose. It can provide reference links to all of the administrative systems that manage the bundle components and oversee the terms and conditions of the bundle.

1.4.3.5 Process Innovation

SOA in general and MDM specifically are enablers of process change and innovation for the enterprise (we will go into more detail on the relationship between SOA and MDM in the next chapter). MDM delivers data management services to the enterprise that closely align with business tasks that manage master data. Therefore, the definitions of these services can be directly used in the conception and process modeling phase of a process innovation project. Additionally, the implementation of these services enables the enterprise to realize the process innovations much more quickly than was previously possible. Previously, process innovations would have resulted in extensive impact on the scattered master data elements in the enterprise. Where does the process retrieve its customer name and address information from?

Where can it find the related products sold to those customers? How are both of these source systems organized so the data can be retrieved effectively? All of these questions and this complexity would have to be dealt with each time an innovation in business process was considered. Because the implementation of the MDM System has already resolved these issues, it is now easier to change existing business processes and support process innovation.

1.4.3.6 Market Innovation

Marketing is focused on creating, winning, and retaining customers. Marketing innovations deal with the identification and development of new ways of achieving this. including new product designs or packaging, new product promotions, or new media messages and pricing. Successful marketing innovations depend heavily on the quality and timeliness of the market and enterprise data they use to do their analysis. This is where an MDM implementation can be very beneficial to these innovation initiatives. The MDM Customer Data Integration system contains the most accurate, up-to-date, and complete view of the current customers, vendors, and prospects. The MDM Product Information Management system contains the most complete view of the products available in the enterprise. Combined with data from other master data domains, this is a wealth of information vital to the marketing analysis and market innovations. The data warehouse is another typical source of data consulted for this purpose. Because cleansed data from the MDM System is an excellent data source for a data warehouse, its usage for marketing innovations is complementary to that of MDM.

1.4.3.7 Supply Chain Innovation

As communications and transportation have rapidly increased, opportunities for change in the supply chain have increased dramatically. The Internet has brought together suppliers and buyers that were previously unaware of each other and has opened world markets for even the smallest organizations. Internet advancements have also opened up the market for labor to be employed where it can be most economically sourced. All of these new possibilities offer opportunities for supply chain innovation. To implement these innovations, enterprises need to optimize their ability to switch between in-house and outsourced parts of their supply chain, cultivating the ability to quickly switch from one supplier to the next or from one distribution model to the next. When master data is not well managed, making such changes can lead to serious business errors. Well-defined and well-managed master data, combined with well run MDM processes, allows for quicker implementation of supply chain innovations. New sources of master data can be incorporated in the MDM infrastructure quickly, and because all of the data will go through the same data quality processes and the same MDM business processes, the overall stability of the corporate master data won't be affected negatively. It is also much easier to supply master data to new elements within the supply chain requesting it. Because of its hub architecture, the number of changes that need to be applied to application interactions is much smaller than in a traditional network infrastructure where unmanaged master data exists in many different systems. In Figure 1.18, the addition of a new distribution channel (A) leads to fewer changes (dotted arrows) in the enterprise application infrastructure when an MDM System is present.

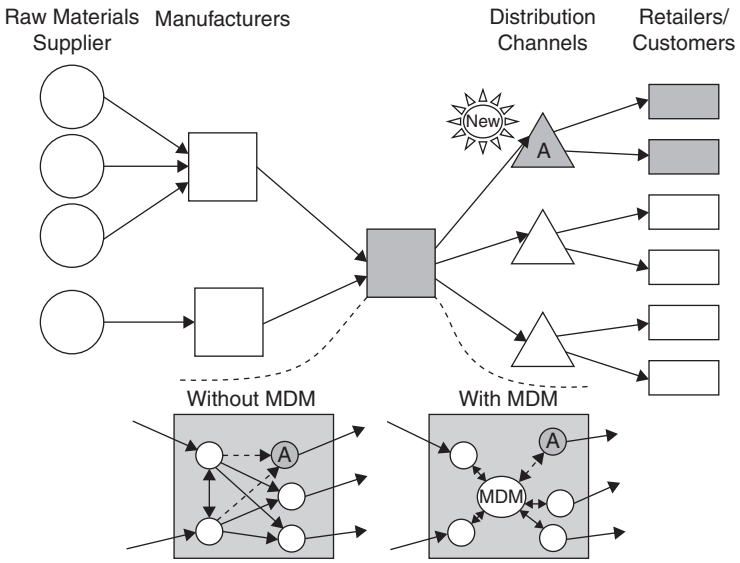

Figure 1.18 Supply Chain with and without MDM.

1.4.3.8 Accommodating Mergers and Acquisitions (M&A)

As we saw earlier, mergers and acquisitions are a common cause of the existence of unmanaged master data in an organization. The successful integration of data is heavily dependent on the data governance practices and the data quality standards of the participating organizations. If data governance is not practiced well within one of the participants, it is going to be very hard to identify the data sets that need to be consolidated. In many cases, much of the data is kept in places that IT is not even aware of, such as in spreadsheets or local databases. If the quality of the data is not adequate, the confidence in the data is low and consolidation is going to be problematic even when adopting an MDM strategy. More often than not, the anticipated cost of such a data consolidation effort is underestimated and partly to blame for a high number of failed mergers and acquisitions. The Boston Consulting Group estimated that more than half of the mergers and acquisitions between 1992 and 2006 actually lowered shareholder value.

1.4.3.9 Introduction of New Requirements

Business changes continuously, not just for the reasons described earlier, but often just to keep up with competitors and stay in business. In the change patterns described in this section, the role of an MDM System is to accommodate business change more easily and more rapidly within the enterprise, and to do so in a more controlled and governed manner.

First, let's consider changes to usage patterns and the user community. The user community that manages and retrieves master data changes over time. Common examples can be found in the many self-service Web sites where customers can change name, address, and phone number information online or through the integration of a vendor portal for a retailer.

These self-service and portal-based usages add whole new communities of end users to the enterprise. Using an MDM System, these new user communities can be integrated into the total user community by assigning them the appropriate security rights and privileges, providing them with suitable user interfaces, and managing the additional workload by scaling up the MDM System to an appropriate level.

Management of master data is governed by many business rules—rules that determine data validity, entity lifecycles, decision-making processes, event handling, matching, and merging. Over time these rules tend to change. Some rules are relaxed, others are tightened, others are corrected, and still more change because of external influences (e.g., regulatory or legislative changes). When using an MDM System, these business rules are encapsulated within services, so service consumers should not need to change their integration logic as rules change. Also, because business rules are componentized inside the MDM System, they are easy to change. Such changes require appropriate governance, as we will describe later in this book.

It is hard to predict future usage patterns of any IT system, and MDM is no exception. Thus, an MDM System must allow for new services to be built that address to new requirements without disturbing any of the existing integrations. It is important that the MDM System is able to quickly support such new services to reduce IT project implementation time. An MDM System typically contains tooling to create new service definitions augmenting the default set of services provided with the product license.

Few products change as frequently and dramatically as software products. In an enterprise context, there is a never-ending stream of new applications, new versions, new releases, and new fix packs of all of the software products being used with the organization. These must fit into the IT infrastructure with minimal interruption to the business. An MDM System can help accommodate some of these types of changes in a number of ways. First, an MDM System must respect backward compatibility. In other words, it must make sure that newer versions of the MDM software, with new and improved functionality, can be introduced without affecting existing integrations with the rest of the IT infrastructure. The new system must support the existing services, and the existing integrations and data, in order to not disrupt the business after upgrade. Secondly, MDM Systems typically run on a stack of other software products such as database management systems (DBMS), application servers, and message middleware. The MDM System isolates the end user of the services from the details of the underlying infrastructure. Clients of the system should not require software upgrades if an underlying stack component is upgraded. Thirdly, MDM Systems typically run on a variation of different hardware platforms. This enables the enterprise to select the platform most suitable for its needs. Infrastructure stack dependencies can become opportunities if the same MDM System can run on Microsoft®, Windows®, and IBM z/OS® Mainframes. Finally, the usage of SOA architecture (see Chapter 2) with service-based interfaces isolates the MDM users and client software from the details of the underlying implementation and related changes.

As we saw earlier in the discussion on mergers and acquisitions, MDM can play an important role in bringing data from the participating companies together into one managed location for master data. But adding additional sets of data is not only related to mergers and

acquisitions. In many cases, enterprise MDM enablement is phased in by addressing only one subset of the applications and their data in the enterprise at a time. After an initial load involving a few enterprise systems, the rest is phased in iteratively over time. Even in relatively mature deployments, batch loads into the MDM Systems are fairly common. Facilitating quick and accurate migrations, creating initial or delta loads is an important capability of an MDM System, allowing the business to leverage the advantages of MDM more rapidly and to react efficiently to changing environments.

Conclusion

MDM is a broad subject that touches on many of the concepts of enterprise information architecture. MDM strives to untangle and simplify the complex systems that have evolved to manage core business information by logically consolidating this information into managed yet flexible MDM Systems. Acting as either a system of record or a system of reference, MDM Systems can provide authoritative data to all enterprise applications.

As we have described throughout the chapter, successful MDM Systems:

- Provide a consistent understanding and trust of master data entities
- Provide mechanisms for consistent use of master data across the organization
- Are designed to accommodate and manage change

These are the key principles of MDM that we will continue to detail throughout the remainder of the book.

The business drivers behind MDM are compelling—from regulatory compliance to improving the responsiveness of an organization to change. By providing authoritative information as a set of services, MDM is also a key enabler for broader enterprise strategies, such as SOA. The following chapter will dive into the details of SOA and the role of MDM Systems in an SOA environment.

References

1. Inmon, W. H. 2005. *Building the Data Warehouse*. New York: John Wiley & Sons.
2. Haas, Lin, and Roth. 2002. Data Integration through Database Federation. *IBM Systems Journal* 41(4):578–596.
3. Gray, J., and Reuter, A. 1993. *Transaction Processing: Concepts and Techniques*. San Mateo, CA: Morgan Kaufmann.
4. Inmon, W. H. 1999. *Building the Operational Data Store*. New York: John Wiley & Sons.
5. Inmon, W. H., O'Neil, B., and Frymann, L. 2007. *Business Metadata: Capturing Enterprise Knowledge*. San Mateo, CA: Morgan Kaufmann.
6. van der Vlist, E. (2002). *XML Schema*. Sebastapool, CA: O'Reilly Media.
7. Gilb, T. *Principles of Software Engineering Management (1988)*: Principles of Software Engineering Management, Addison-Wesley Professional.

8. Eckerson, W. W. 2002. *Data Quality and the Bottom Line, Achieving business success through a commitment to high quality data*. Retrieved 03/17/2008 from The Data Warehousing Institute at http://www.dw-institute.com/display.aspx?id=6045.
9. USA Comptroller of the Currency. *Bank Secrecy Act*. Retrieved 03/17/2008 from http://www.occ.treas.gov/handbook/bsa.pdf.
10. Bank for International Settlements. *History of the Basel Committee and its Membership*. Retrieved 03/17/2008 from http://www.bis.org/bcbs/history.pdf.
11. Lickel, C. W. 2007. Introduction. *IBM Systems Journal* 46(2).
12. Wayman, R. J. 2005. *The Trickle Down of SOX*. Retrieved 03/17/2008 from ResearchStock.com at http://www.researchstock.com/cgi-bin/rview.cgi?c=bulls&rsrc=RC-20050311-.
13. Davila, T., Epstein, M. J., and Shelton, R. 2006. *Making Innovation Work: How to Manage It, Measure It, and Profit from It*. Upper Saddle River, NJ: Wharton School Publishing.
14. Amabile, T., Conti, R., Coon, H., et al. 1996. Assessing the work environment for creativity. *Academy of Management Journal* 39(5): 1154–1184.

Index

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z