

Delivering information you can trust

June 2007



IBM **Information Management** software

IBM Multiform Master Data Management: The evolution of MDM applications

Contents

- 2 Traditional approaches to master data management**
 - 2 The enterprise application*
 - 4 The data warehouse*
 - 5 Integration middleware*
 - 6 The problem with traditional approaches*
- 7 The evolution of master data management solutions**
- 10 IBM Multiform Master Data Management**
 - 10 The value of IBM Multiform Master Data Management*
- 13 Alternatives to multiform MDM**
 - 13 Index MDM and reference MDM*
 - 14 Immature MDM approaches*
 - 15 Uniform MDM*
- 16 Evolutionary chasms associated with multiform MDM alternatives**
 - 16 Data-centric versus function-centric approaches*
 - 18 Uniform versus multiform MDM applications*
- 20 Overcoming evolutionary chasms with IBM Multiform MDM**
 - 20 Scalability*
 - 21 Total cost of ownership*
 - 21 Flexibility*
 - 22 Time and risk*
- 22 IBM Multiform MDM: The true strategic MDM solution**

Companies across many industries face business challenges that affect their master data—the high-value, business-critical information about customers, suppliers, products and accounts—and the ability of IT to deliver on the requirements of a dynamic business. This critical business information is replicated and fragmented across business units, geographic branches and applications. Enterprises now recognize that these symptoms indicate a lack of effective and complete management of master data.

Since companies began shifting from a mainframe-based architecture to a more flexible distributed architecture, IT departments have attempted to gain control over this master data using a variety of methods. But few have demonstrated true success due to their reliance on existing, but repurposed, systems and applications.

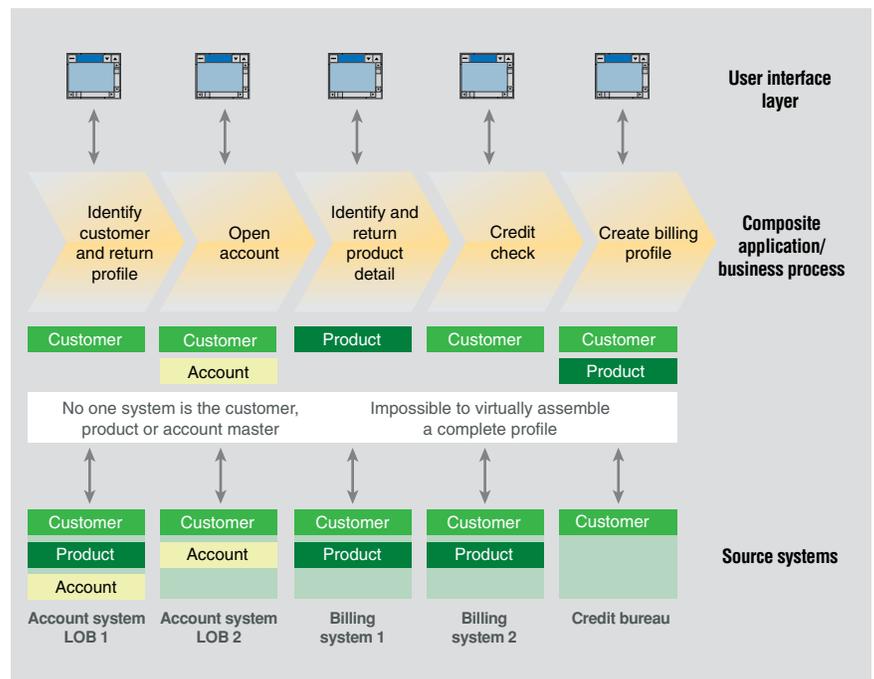
Traditional approaches to master data management

The enterprise application

Traditional approaches to master data include the use of existing enterprise applications, data warehouses and even middleware. Some organizations approach the master data issue by leveraging dominant and seemingly domain-centric applications, such as a customer relationship management (CRM) application for the customer domain or an enterprise resource planning (ERP) application for the product domain. However, CRM and ERP, among other enterprise applications, have been designed and implemented to automate specific business processes such as customer on-boarding, procure-to-pay and order-to-cash—not to manage data across these processes. The result is that a specific data domain, such as customer or product, may actually reside within multiple processes, and therefore multiple applications.

In this scenario using application masters, it is difficult to determine which iteration of customer, product or account—if any—is complete and correct (see Figure 1). Additional complexity occurs as organizations attempt to maintain the correct copy of the data, and identify and understand all of the systems that can update a particular domain, those that consume portions of the updates, and the frequency rate at which this consumption occurs. It quickly becomes apparent to organizations that have undergone such a project that the process-automating application cannot also manage data across the enterprise.

Figure 1: Serving business processes without master data management



The data warehouse

Alternately, some enterprise initiatives have attempted to repurpose new or existing data warehouses to serve as a master data repository. As data warehouses aggregate enterprise information, the warehouse is often viewed as a starting point for companies attempting to master their data. However, data warehouses have inherent design characteristics to optimize reporting and analysis, and to drive sophisticated insight to the business. This design, while effective for its primary use, cannot scale well within an operational environment—even in the case of dynamic warehousing—when measured against the needs of most businesses today.

Based on its fundamental design, the data warehouse also lacks data management capabilities. Essential functionality such as operational business services, collaborative workflows and real-time analytics that are critical to success in these types of master data implementations require large amounts of custom coding. Similarly, data management capabilities—data changes that trigger events and intelligent understanding of unique views required by consuming systems—are also absent from a data warehouse.

Integration middleware

Enterprise information integration (EII) or enterprise application integration (EAI) technologies used to federate and synchronize systems and data have also been presented as substitutes for data management products. Although these solutions can tie together disparate pieces of architecture either at the data tier (EII) or at the application tier (EAI), they do not provide either a physical or virtual repository to manage these key data elements. And much like warehouses, they lack data functionality.

The management of data processes poses yet another challenge. Choosing to build functionality within this middleware technology can affect performance in its core competency: the integration of applications and data. Without a true master data solution to complement it, the implementation of EII and EAI technology can actually add to the architectural complexity of the business and perpetuate master data problems with point-to-point integration.

In most cases, these methods fail because they are designed to treat data symptoms, such as fragmented data or systems that are out of sync, not the root cause of the master data problem. That root cause is that data is tightly coupled to applications and business processes and this data is not being managed by a single, independent resource that can capture the complete and current enterprise understanding of the domain (customer, product, account or supplier).

While EII and EAI technologies specialize in specific functions such as data federation, data quality or aggregate analytics, they do not manage the essential data processes or data changes that can initiate other processes such as quality and data stewardship. Attempting to manage these data processes virtually can mean that an essential fact—like the correct address of a customer—must be determined on every transaction; for example, determining whether address 1 from system A or address 2 from system B is correct. It is also necessary to persist this information because the data is created and changed over time—this timeframe is known as the information lifecycle—to capture net new data like privacy preferences and to deliver this information in context to all of the relevant consumers, typically on demand via business services.

The problem with traditional approaches

The following example illustrates the problem. A customer contact occurs in the call center. This action initiates an address change to a customer record. The address change is immediately reflected in the CRM application, but the billing system is not updated. The customer's bill for that month is sent to the wrong address and the analytics are skewed because the data warehouse did not receive the required change. The ERP system, on the other hand, has a third address, confusing data stewards and forcing another customer contact to try to correct the error. The result is a poor customer service experience for the customer.

No single application has the ability to manage the “golden copy” of this customer information to ensure all systems receive the necessary changes, as well as triggering duplicate suspect processing (matching the customer with

an already existing address), event handling (such as alerting a data steward to the problem) and analyzing whether a product offer should be made due to the change. While existing systems are automating their associated business processes, this dynamic data is actually driving process changes of its own.

Integration technology or a data warehouse in combination with extensive customization may provide the ability to link some of these applications and data elements. But does this integration occur frequently enough to avoid discrepancies across the enterprise? What if the address change was originally made to the billing system when the customer received the last invoice statement? Will this information be overwritten by the dated CRM address? What happens with the addition of another channel such as a self-service Web application that also has an address update capability?

The evolution of master data management solutions

In general, master data management (MDM) solutions should offer the following:

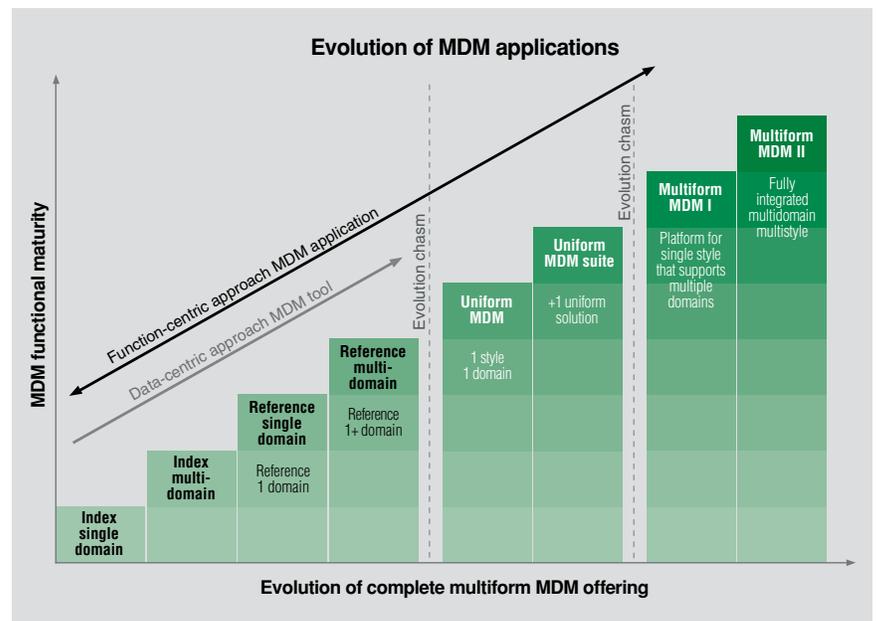
- *Consolidate data locked within the native systems and applications*
- *Manage common data and common data processes independently with functionality for use in business processes*
- *Trigger business processes that originate from data change*
- *Provide a single understanding of the domain—customer, product, account, location—for the enterprise*

MDM products, however, address these four requirements very differently. Some products decouple data linked to source systems so they can dynamically create a virtual view of the domain, while others include the additional ability to physically store master data and persist and propagate this information. Some products are not designed for a specific usage style, while others provide a single usage of this master data. Even more mature products provide all of the usage types required in today's complex business—collaborative, operational and analytic—as out-of-the-box functionality. These mature products also provide intelligent data management by recognizing changes in the information and triggering additional processes as necessary.

Finally, MDM products vary in their domain coverage, ranging from specializing in a single domain such as customer or product to spanning multiple and integrated domains. Those that span multiple domains help to harness not only the value of the *domain*, but also the value *between* domains, also known as relationships. Relationships may include customers to their locations, to their accounts or to products they have purchased. This combination of multiple domains, multiple usage styles and the full set of

capabilities between creating a virtual view and performance in a transactional environment is known as *multiform master data management*. Figure 2 depicts these different solutions and their placement in functional maturity versus MDM evolutionary stages.

Figure 2: Evolution of MDM applications



IBM Multiform Master Data Management

IBM® Multiform Master Data Management manages master data domains—customers, accounts, products—that have significant impact on the most important business processes and realize the promise of Service Oriented Architecture (SOA). IBM is the only vendor that delivers an integrated MDM product with significant out-of-the-box functionality for each MDM usage style—collaborative, operational and analytical—across multiple data domains, thereby managing the complete data lifecycle.

This multiform approach enables an organization to centralize its most critical information to a single trusted source and provide configurable functionality across multiple usage types and data domains, which can be altered to unique business requirements. Multiform MDM delivers capabilities such as identifying the most valuable customers, introducing new products and product bundles more quickly and managing threat and fraud risk more effectively.

The value of IBM Multiform Master Data Management

The value of IBM Multiform Master Data Management can be recognized in a range of projects, from short-term fixes for a narrow set of problems such as capturing customer privacy preferences to long-term enterprise-wide initiatives like delivering agility to the business by shifting to SOA.

IBM is committed to providing flexibility to its customers; therefore, IBM Multiform MDM is designed to scale from tactical requirements to full-blown strategic solutions with enhanced value through the understanding of domain relationships. Customer-related examples of tactical requirements to strategic initiatives that realize the value of Multiform MDM include:

Identifying the most valued customers. The tactical requirement is to meet short deadlines and deliver quick return on investment (ROI). IBM Multiform Master Data Management (customer domain/operational usage) is deployed in a single line of business to consolidate customers across multiple CRM and billing applications. It also provides an operational view through the use of business services to the customer-facing channels of the business. While the tactical problem may be solved with a number of MDM solutions across the MDM maturity model, relatively immature solutions create future risk by inhibiting an organization's ability to add a second domain (account) and to integrate new applications to the MDM product. However, using multiform MDM protects the organization from a strategic standpoint over time, as business requirements expand to provide offers to customers based on accounts owned (bundling) across multiple lines of business. The IBM Multiform MDM operational/customer component is integrated with the IBM Multiform MDM operational/account component, spanning multiple lines of business and allowing the organization to provide differentiated service to its customers and leverage additional revenue opportunities.

Introducing new products to market more efficiently. Organizations need to streamline the new product introduction (NPI) process—the tactical requirement. Using IBM Multiform MDM (product/collaborative), companies can create a single repository for products across the enterprise, quickly leverage configured existing workflows, and create and augment product data spanning the process. This process helps shorten the NPI process from weeks to days. Again in this scenario, the organization may choose an immature MDM product to create this efficiency, but create risk in the ability to add a second domain (supplier) and another usage style (operational) needed for both the product and supplier domains. Strategically, the organization can integrate the IBM Multiform MDM product/collaborative component with a supplier/operational component to update transactional systems via business services with products available from specific suppliers, reducing stock outages and providing real-time supplier alternatives.

Organizations moving from tactical to strategic deployments of MDM—Phase 1 to Phase 2 and beyond—can face potential risks, or chasms, if existing functionality and domain focus is not available. Caused by heavy customization, high development costs and product scalability limitations, chasms can prevent initiatives from reaching their true value of leveraging MDM usage styles and combining interrelated domains. IBM Multiform MDM provides this value from both usage and domain perspectives to help ensure organizations can map their MDM capabilities to their ever-changing business needs.

Alternatives to multiform MDM

Alternatives to multiform MDM include indexing and reference technology as well as uniform master data management.

Index MDM and reference MDM

Index MDM provides a global ID linking repository for master data domains, creating a virtual hub. Rather than physically retaining data, the index cross-references those systems that contain fragments of a data domain and dynamically creates a view for the user, upon request.

Reference MDM, the next step on the MDM evolutionary ladder, often physically stores a small subset of master data in addition to global IDs for consumption. Much like the index MDM, reference MDM is designed to deliver a read-only view of the data domain and requires custom-built business services. Limitations to the accuracy of the data may exist because of update scheduling, often conducted in a batch process.

Both index and reference MDM approaches are data-centric and tool-based because they provide sets of tools or templates to build components such as a data model and business services. Because of the narrow tactical focus of the initial phases in many MDM projects, index and reference data can provide a solution to such problems. As organizations evolve into secondary phases, deployments usually have a need for not only index- and reference-type data, but also for managed master data with functionality.

Immature MDM approaches

Revisiting the address change example, departments (other than the contact center) may now have the ability to see two different addresses: a CRM address (remember this was recently updated) and another address in the billing system. However, they cannot determine which address is correct or conduct an automatic update to these consuming systems with the most current version of the truth.

In these situations, MDM project teams can become paralyzed by the inability to scale, overwhelmed by customization requirements and challenged by a lack of essential functionality. This can lead to much higher total cost of ownership (TCO) for index and reference toolsets, which appeared to be much more affordable than a more mature MDM product from a software licensing perspective. This technological, functional and cost barrier to increasing MDM scope is known as the data/function evolutionary chasm.

Uniform MDM

The next stage in the evolution of MDM, an enhancement in functional maturity from a reference product, is uniform master data management. Uniform MDM maps a single form of usage to a single primary domain; for example, product with a collaborative usage style. Uniform technology recognizes differing usage styles of master data across an enterprise, building in additional functionality determined by the vendor to be the most dominant use of that data. Because these products provide a solution for only a single use of a particular domain, a second and more considerable chasm lies between the MDM application types: the integrated usage chasm that results from the independent silos of master data.

Although in theory, these master data silos could be fused through middleware, practice has shown that building the necessary functionality within the middleware layer can overload these technologies and affect performance, formulating a similar result as a pure middleware solution to the MDM problem, as discussed earlier. The uniform MDM suite has the same set of capabilities and limitations as uniform MDM, but it is a collection of these disparate, single-domain, single-usage MDM products often created or based upon proprietary middleware or applications. While uniform MDM suites provide domain or usage selection (again, a single usage is tied to a particular domain), these products are not integrated to unlock the value among domain relationships.

Evolutionary chasms associated with multiform MDM alternatives

Evolutionary chasms can occur with index, reference and uniform MDM: the data/function and integrated usage chasms.

Data-centric versus function-centric approaches

Organizations that choose to move forward with data-centric technology limited to the capability of either an index- or a reference-based MDM product run serious risks of falling into the data/function evolutionary chasm.

These risks occur because of the attempts over time to expand the scope of MDM and encompass future needs of the business. The chasm represents the technological and functional barriers of moving from a tool-based data approach (index and reference) to an application-based function approach (uniform and multiform). This chasm presents challenges such as:

Tooling rather than immediate functionality. Rather than provide an MDM application complete with out-of-the-box functionality, products focused solely on an index or reference approach to MDM generally provide a set of tools to build MDM components such as a custom data model. At first glance, this appears to offer flexibility, but it can result in high costs and time-consuming implementations associated with deployment because the organization is faced with extensive customization.

Instead of offering an extensible model with configuration capability, index and reference styles force organizations to fully understand and plan for all current and future business requirements before embarking on an MDM initiative. Failure to plan for future requirements can result in unanticipated flaws in the model and mandate a complete redesign of the MDM solution, significantly increasing development and integration costs.

This need to redesign a legacy solution originally built to master data is commonly seen in, but not exclusive to, the financial services industry. In the past, these companies, through the use of tooling, built client files that did not have inherent flexibility to adapt to the needs of today's environment, such as the need to capture privacy preferences, add new channels (such as ATMs or the Web) and new products.

Pursuing a data-centric approach necessitates heavy customization, which in turn affects the ability to seamlessly upgrade the product when necessary, and forces the organization to bear the burden of ensuring that the development of components such as the model and business services is repeatable. Instead of demonstrating expedient value through the reuse of MDM components, development of new components in subsequent phases can also extend deployments dramatically, often derailing projects permanently.

Single domains and lack of usage styles. Toolsets and templates available in data-centric MDM technology are typically designed for a single domain; that is, additional MDM products—likely supplied by different vendors—would be required as organizations move to subsequent phases in their MDM deployment. The result could be additional integration costs, poor performance and slower time to value. In addition, data-centric approaches lack functionality, and any style of usage (collaborative processes, operational services or real-time analytics) must be created from the ground up.

Uniform versus multiform MDM applications

Even with function-centric MDM applications, organizations can face a steep evolutionary chasm that inhibits their ability to add additional usage to a primary domain or leverage the relationships between domains. This integrated usage chasm can block organizations with uniform MDM from evolving to a multiform MDM deployment for several reasons:

- ***Single usage styles.*** *Uniform MDM forces an organization to choose between usage styles and domains. Because vendors of this type of MDM provide separate and distinct platforms for usages across domains (for example, an analytical platform for the customer domain and a separate operational platform for product domain), clients are mandated to choose the domain most critical to their business at the current time and trust that they will have the ability to custom-integrate these platforms as needs grow across domains or across usage styles.*

- **Proprietary middleware and applications with the uniform suite.** *The uniform MDM suite is typically offered by application suite vendors using data models evolved from the applications themselves. This approach affects the suitability of the product in heterogeneous environments and dictates prior middleware purchases and subsequent application investments.*
- **Value locked up between domains.** *Uniform MDM does not provide an integrated platform for a usage style across domains, creating silos of master data and preventing an understanding of relationships between domains. Fusing the silos using middleware does not overcome this problem.*

Consider the following example. A customer purchases a product, which creates or changes data within the MDM customer domain application and also in the separate MDM product domain application. Each MDM application persists the information most applicable to its own domain focus, but that information is not integrated with the other domain, in effect creating silos of master data. Because the customer and product domains have common intersection points (such as the customer's discount or eligibility for a particular product), middleware is introduced to fuse together the two MDM applications. Unfortunately since middleware is not designed to manage the data itself, it creates a performance bottleneck. Again, this forces the organization to manually manage the data integrity across multiple applications.

Uniform MDM does not provide the integration necessary to develop a comprehensive view and understanding of customers, the accounts they hold, the products they have purchased and the delivery locations.

Overcoming evolutionary chasms with IBM Multiform MDM

With several chasms in the MDM evolutionary chain, carefully planned strategic decisions about master data management projects, regardless of initial scope, can make a difference in the success of master data management implementations. IBM Multiform MDM helps ensure that organizations can meet tactical timelines and deliverables, while providing the stability of an expandable solution designed to grow with the requirements of the business.

Scalability

IBM Multiform MDM addresses all of the stages throughout the MDM evolution. It enables data-centric tooling that may be necessary in initial-phase deployments while providing the roadmap to increase the scope over time—such as managing data for new businesses or projects. Multiform MDM is built to allow organizations to easily include additional domains and usage types without the use of customized integration that is needed with all other MDM types. IBM Multiform MDM, a superset of all steps in MDM maturity, can be phased in over time to meet the unique needs of the business.

Total cost of ownership

During the evaluation of MDM products, organizations often focus on cost from a licensing perspective. However, the nature of MDM dictates an evaluation for TCO based on several parameters, including planning, integration with new and existing systems, development needed for the initial functionality, support and more. IBM Multiform MDM is designed to offer the lowest TCO option available on the market today, with out-of-the-box functionality spanning all usage types and capability to manage customer, product, account and location as well as other primary and secondary domains.

Flexibility

IBM Multiform MDM offers extensive flexibility compared to other MDM products. Unlike uniform MDM, IBM Multiform MDM leverages open standards such as Java™ 2 Platform, Enterprise Edition (J2EE), XML, Web services and more. Further, fine- and coarse-grained exposed business services assure organizations that IBM Multiform MDM is configurable to unique business processes, rather than mandating changes to applications and processes to accommodate an MDM product.

Time and risk

Time to value and risk are critical aspects in the selection of a MDM product. IBM Multiform MDM not only can be componentized based on project requirements, but it also provides predefined integration points to complementary technologies such as EAI and EII (such as IBM Information Server), as well as third-party reference databases and data quality tools. This packaged integration, coupled with robust extensible data models, can significantly shorten implementation time and improve time to value while minimizing risk.

IBM Multiform MDM: The true strategic MDM solution

While companies today are often entering today's MDM market with tactical requirements, it is important to account for the long-term strategic implications of decisions regarding MDM application infrastructure. While providing significant value to the business, IBM Multiform MDM can help mitigate the risks of this critical decision and provide a componentized solution that helps organizations eliminate the threats presented by MDM evolutionary chasms.

For more information

For more information about IBM Multiform Master Data Management, please contact your IBM marketing representative or IBM Business Partner, or visit

ibm.com/software/data/masterdata



© Copyright IBM Corporation 2007

IBM Software Group
Route 100
Somers, NY 10589

Produced in the United States of America
June 2007
All Rights Reserved

IBM and the IBM logo are registered trademarks of International Business Machines Corporation in the United States, other countries or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc., in the United States, other countries or both.

Other company, product and service names may be trademarks or service marks of others.

References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

TAKE BACK CONTROL WITH **Information Management**