The SOA Magazine

Feature Article




Data Management: The Missing Link in Your SOA Strategy

by David Butler and Jeff Pollock

Published: July 18, 2008 (SOA Magazine Issue XX: July 2008)

Download this article as a PDF document.

Abstract: SOA is transforming the IT landscape and redefining the way businesses integrate software applications. Properly deployed and managed, it empowers business flexibility where there was none. But SOA is only as good as the data it leverages. Bad data has ruined too many enterprise software projects. Enter data management.

Data Management encompasses data integration together with master data management (MDM) which addresses the governance of data-centric environments. MDM aims to improve business data quality while providing a single, unified view of that data.

How can an SOA leverage such a single view? The bridge that ties these seemingly disparate paradigms together is data integration, which combines the data-centric elements of both SOA for data services and MDM foundational techniques for data quality, data profiling, and data relationship management. When used together in this way, organizations can reap a sizeable and sustainable competitive advantage as a result of flexible information architectures and flexible authoritative data.

Introduction: The CIO Conundrum

You're the CIO and you are asked to do more with less and be more accountable at the same time. Is it even possible? While SOA is a necessary part of the solving the CIO conundrum, it is not enough. SOA alone simply does not address the issues of data management, data quality, data architecture or data modeling.

Master data management is a modern technology framework designed to eliminate poor data quality in IT systems. It employs tools for managing data quality, data cleansing, data relationships, and data lineage which acts to look at the survivorship of data throughout the lifecycle. Together with Data integration, MDM can connect data marts and data warehouses through data synchronization and offer the ability to move data in addition to simply managing it.

MDM and data integration together complete the SOA vision by enabling high-quality implementations that go beyond loosely coupled plumbing in order to address the complexities of data. This combination has a profound impact on business operations. Ultimately, it begins to answer the CIO conundrum by reducing costs at the source, while at the same time providing benefits of increased information agility. Let's look at this in detail.

Enterprise Business Processes

Business processes are the operational embodiment of everything the organization does. Automating these business processes is what operational IT systems do. Efficiency on this front is critical for a company's survival. Each department and every line of business has its own processes, many of which can be supported by a single application or set of unified applications as part of a suite. But the most strategic and complex business processes cross applications and departments.

These types of processes often become trouble spots that are ripe for SOA and MDM based solutions. Because of the cross-application and cross-departmental nature of these systems, the efficiency and flexibility of the data, transactions, security, and management is at a premium.

To illustrate how SOA and MDM can work together to optimize such processes we'll be exploring a simplified Order-to-Cash process throughout this article. Three lines of business are supported by three different applications: Sales, Order Management, and Accounts Receivable. Each department has its own applications and manages its own customer data.

- *Sales* - A sales team is supported by a front office sales automation application that automates opportunity management with point-of-sale support. Order capture information is collected on price and discounts. Contracts are established with customers. Customer information is maintained and governed locally according to department rules.
- *Order Management* - Order management is supported by a back-office application that manages the configuration of products sold and shipping. It enables customer access to order status and supports a call center. Customer information is maintained and governed locally according to order management department rules.
- *Accounts Receivable* - Accounts receivable is supported by a financial application that automates invoice generation, billing, and if necessary, collections. Customer information is maintained and governed locally according to financial department rules.

The Cross-Departmental Order-to-Cash Process

Order-to-cash is the business process that starts when an opportunity, turns into a sale, and ends when the money is in the bank. This process crosses the sales, operations, and accounts departments and their supporting application software. Inconsistent data in these application silos can severely impact these processes. Poor quality data that might be good enough to support the local operations of each organization can break the enterprise processes that need to span several departments.

Figure 1 illustrates a simple example of how customer data can vary.

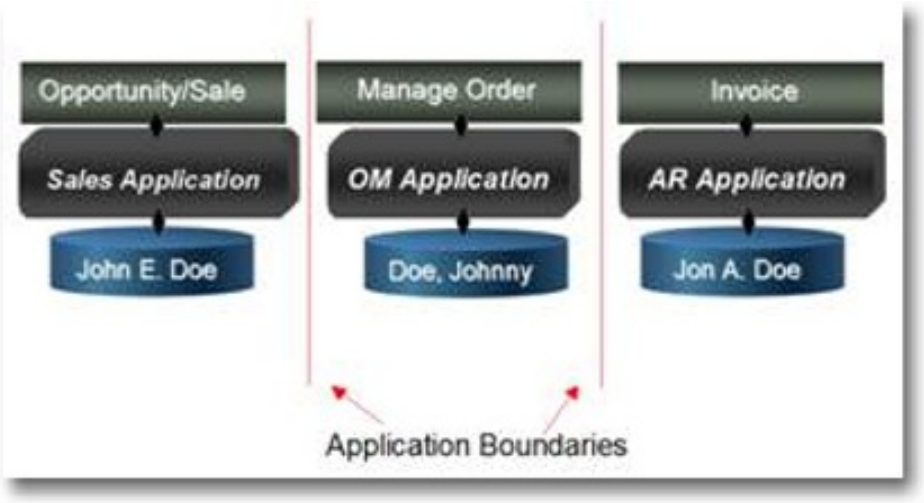


Figure 1

Each application has its own version of our customer John Doe. In this simple example, we can imagine a simple misspelling of the customer's name, but other differences may include address typos, or out-of-date records, missing unique identifiers, incorrect customer attributes, misaligned product codes, missing summary data for the customer or the customers household, etc. If these data differences are not correctly dealt with at the application boundaries, the business process breaks down and even though the messages may continue to flow, the bad data causes the purported value of this enterprise process to fail. Whether the product is sent to the wrong address, the bill is sent to the wrong person, or the process winds up in a slow and costly manual procedure, to correct the errors, the business operation is left to suffer the consequences of bad data and partially effective systems integration.

Building MDM and SOA Processes Together

One of the primary benefits of investing in a service-oriented architecture is that it enables you to effectively integrate applications and orchestrate business processes. Both of these SOA use cases apply to our order-to-cash process example. Likewise, the MDM use cases apply to the business process because MDM can provide a common system of record for the data about customers, orders, products and accounts. SOA and MDM are natural complements because each provides a layer of decoupling for simplified management of complex systems. For SOA, this decoupling occurs at the messaging layer, whereas MDM decouples complex data and metadata.

Integrating the applications with BPEL and ESB

One best practice for integrating applications together is to provide an enterprise service bus (ESB) that utilizes a shared publish-subscribe framework for high-volume low-latency messaging between applications. The ESB is typically optimized, stateless, and aware of SOA endpoints and their canonical message formats. Typically, each application is connected to the ESB via an API layer (not the database) and the bus eliminates the need for a point-to-point messaging architecture.

An ESB may work alone or alongside other SOA-centric systems such as a business activity monitor (BAM); standards-based data transformation tools (XSLT, XQuery); adaptors to connect the applications; Web services management (WSM); identity management (IdM), a business rules engine (BRE) to control the logic flow, and other event-driven real-time activities.

Another emerging best practice for integrating applications is to leverage a business process engine for orchestrating long-lived and multi-step transactions. Web Services Business Process Execution Language (WS-BPEL) engines are a powerful technology that can execute enterprise business processes from Web services hosted within an ESB, thereby exposing application processes and functions as part of enterprise-scale processes (instead of disconnected endpoints).

Figure 2 illustrates how an order-to-cash process might be built across sales, order management, and accounts receivable domains.

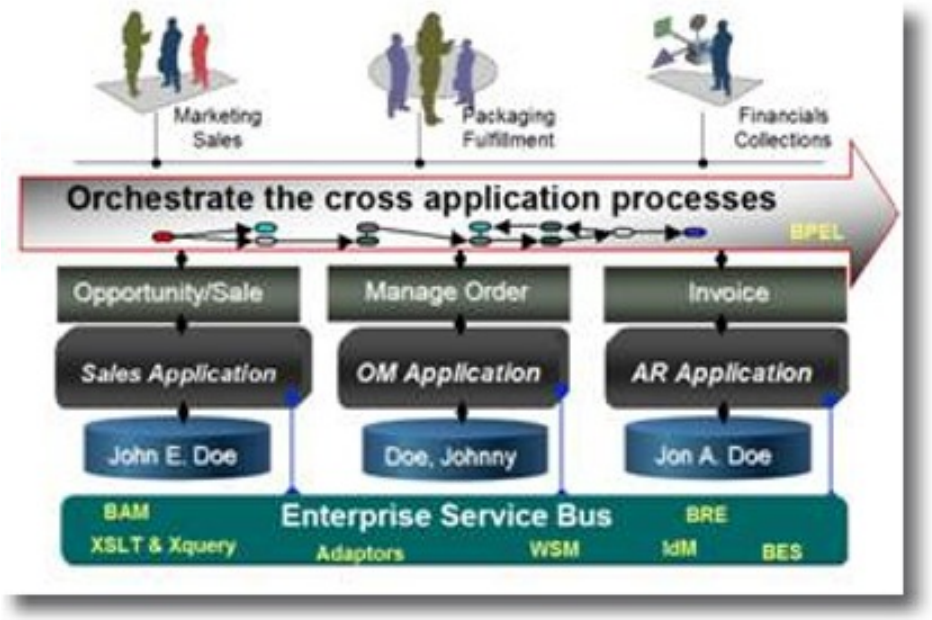


Figure 2

While the ESB facilitates the movement of data between the applications to support the business processes and the WS-BPEL orchestration engine enables multi-step transactions across departments, neither technology addresses internal application data problems. Good ESB and WS-BPEL integration technologies have the necessary technology to help with data fragmentation but are not sufficient to eliminate it.

Without data integration and MDM to fix these data problems, products may still be sent to the wrong address, bills might be sent to the wrong person, and those newly automated WS-BPEL processes continue depend on costly and

time consuming manual data correction procedures.

MDM is SOA for Data

MDM is designed to decouple master data from applications and align it with a centrally governed data model. MDM applications are concerned with the lifecycle and accessibility of high-quality data. They are essentially an enterprise-standard means of managing the data records and keeping them aligned with canonical enterprise models. Once the data is aligned, MDM applications can further help standardize the data, find and eliminate duplicate bodies of data, enrich the data with third-party content; and finally disseminate the authoritative “golden record” that represents a single version of the truth about each master data entity.

In the order-to-cash example, MDM should be responsible for aligning the customer data from sales, order management and accounts receivable applications. Data custodians use MDM applications to define and enforce the corporate data governance rules established by an enterprise governance committee. For this example, MDM can determine that “John E. Doe” from the sales application, “Doe, Johnny” from the order management application, and “Jon A. Doe” from the accounts receivable application are actually one and the same (Figure 3). MDM allows for automatic and manual corrections to any profile information and may also augments the record with third-party demographic data.

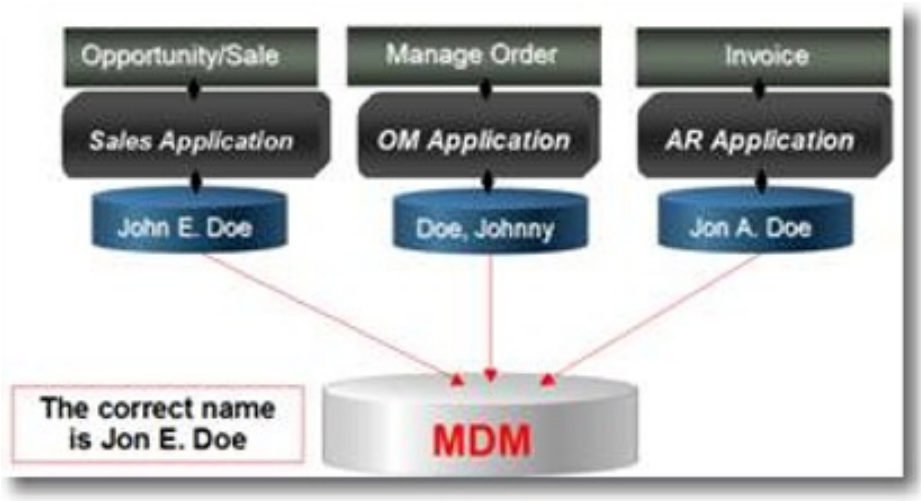


Figure 3

This newly established master data can be used to improve analytic and operational processes. But without SOA, MDM runs the risk of becoming a data silo unto itself. Maximum value for MDM is achieved together with data integration through which its high-quality master data can be placed back into the operational transactional applications that need it to support local and cross-application processes.

Thus, while MDM and data integration provide the common “bus” for enterprise data, it still depends on an SOA infrastructure to get connected to the enterprise integration fabric. Data integration becomes the common denominator for both to be successful.

Deploying MDM as a Service

“MDM as a Service” means that, at minimum, the key features of MDM are addressable as part of an SOA infrastructure. Client applications on an ESB or WS-BPEL process must be able to call MDM as part of their transactions. This simple but powerful design enables a regular SOA transaction to enrich and replace bad data with good data – effectively making the SOA infrastructure itself a trusted source of information. It essentially allows the ESB and WS-BPEL engines to become a meaningful part of the enterprise data architecture.

In our example, the ESB and WS-BPEL engines may be responsible for routing and transforming thousands of transactions per hour. Each transaction may be a particular step in a complex process and may contain a large and complex message.

Let’s take a closer look at the order management system. Its responsibility is to ensure the proper configuration of

products as well as their fulfillment. The service bus will connect the order management system to front office systems (like the sales system) and to back-office systems (like receivables). During the lifecycle of a single transaction (let's say an order booking) the WS-BPEL process may include an MDM enrichment step to add good data to the transaction or an MDM validation step to validate existing transaction data against known good data.

With enterprise MDM services in place, the order management application may only need to receive minimal information about the customer, such as a unique customer ID. The fulfillment processes can then check with the MDM service for the most recent customer shipping and billing addresses. Since the sales application and the order management system use the same golden records, there won't be any mismatch between the two.

Likewise, even if the transactions need to send fully qualified messages that contain all the customer information or product information, the MDM services can behave as an inline validation process to perform record-level matching and raise exceptions in the process before customers are accidentally shipped the wrong products or products are accidentally shipped to the wrong address.

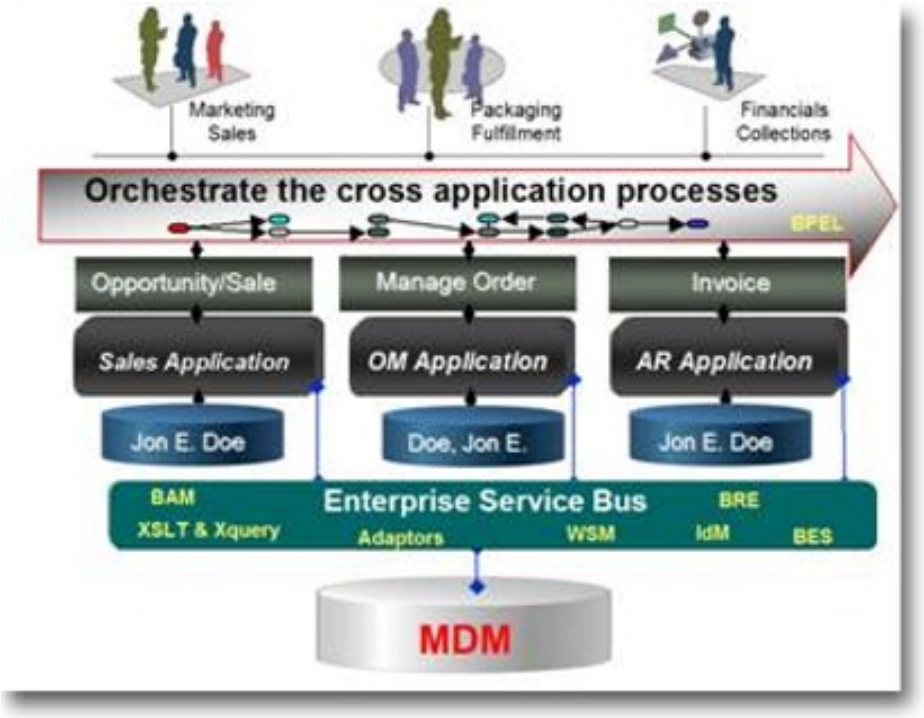


Figure 4

More advanced use cases for "MDM as a Service" might include leveraging MDM for the management of canonical message formats and XSD attribution. Every large SOA infrastructure has hundreds of XML message schemas, but few have a controlled lifecycle for managing schema attributes, naming conventions, dependencies and versions.

As an enterprise SOA infrastructures becomes larger, more complex ,and mission-critical it will begin encountering the same kinds of data, model and metadata management problems that data-intensive systems had to deal with decades ago. Modern MDM applications are precisely designed to solve these kinds of enterprise-scale issues.

"MDM as a Service" will further evolve to offer design-time data governance support for XML schema and message types, thereby enabling developers and designers to keep SOA data in synch with database schema changes and other infrastructure updates.

Let's say that our example sales application reorganizes the compensation territories from being purely geographically-oriented (by region) to being revenue-driven (by account size) as well. With an MDM service in place, the developers could modify the appropriate XML messages to account for these changes (probably by adding new XML attributes to existing messages), increment the version of corresponding schemas, place an effective date by which the new sales hierarchies become effective, and then send the changes to management for approval.

Finally, the rise of grid technology for SOA signals an ideal deployment architecture for MDM services. Instead of having MDM golden records and schema attribution entirely within a database, grid technologies can act as a proxy location to speed the enrichment and validation steps required by high-volume SOA transactions.

Essentially, the ESB or WS-BPEL processes themselves may host a copy of the MDM golden data objects in-memory as part of a grid. Clearly the true master data records would reside in a physical database for persistence, but the runtime execution could be driven entirely from the grid. This advanced architectural style for MDM deployments may not be required for all systems, but for high-performance demands, it could shave critical milliseconds off the MDM enrichment and validation steps during a given transaction.

Conclusion

This article's order-to-cash example has illustrated how SOA and MDM through data integration can be combined. "MDM as a Service" use cases can offer both simple and more advanced places to start:

- MDM as a Web service for SOA client applications to fetch and update master data.
- MDM as a Web service for automatically enriching inline SOA messages with good quality data.
- MDM as a canonical model management layer for SOA XSD definitions.
- MDM as a memory-resident data-grid for state-full ESB transaction management.
- MDM as a callable sub-component within WS-BPEL transactions.

It should be clear that the business value of SOA will increase proportionally as the numbers of MDM-aware services become available. As SOA and MDM technologies become unified through data integration, they can become more widespread and be used together in a multitude of reusable combinations for information agility.

The conundrum that CIOs face with continually dwindling budgets for core business innovation may begin to reverse. By solving the problems of inflexible integration and data rigidity, we can look toward a future where IT strongly drives the center of business innovation.

THE PRENTICE HALL SERVICE-ORIENTED COMPUTING SERIES FROM THOMAS ERL



Copyright © 2006-2008 SOA Systems Inc.