Master Data Management

Pavan Kumar
Memorial University of Newfoundland

Master Data Management (MDM) – Strategies, Architecture and Synchronisation Techniques

Pavan Kumar Purohit

B9021 – Data Management

March 31, 2014

Table of Contents

1. Abstract

In this term paper the author first introduces the concepts of Master Data Management (MDM), Master Data, Data Domain, Customer Data Integration (CDI), Product Data Management (PDM) and One Master Data. Next a business case in support of MDM is presented. In the business case studies various industry scenarios that would require or benefit from a MDM initiative. The implementation of Master Data Management requires business initiative and an IT initiative. The paper will therefore explain various implementation architecture and management framework for MDM implementations that are published in journals and books. The most important part of MDM is data synchronisation techniques. The data synchronisation is required to maintain the integrity of Master Data in a steady state scenario. The paper will explain data synchronisation techniques that could be used. In the conclusion the paper will provide a MDM implementation solution using a case study which will use the concepts explained in the paper. The problem statement in the case study is derived from the authors work experience. In order to complete the term paper multiple articles from various management and technology journal and books were reviewed. These articles are listed in the references table..

2. Introduction

The increasing amount of data is creating challenges to companies' data management practices, causing data quality problems which are very common in today's companies. Additionally today's technology allows storing more data than a company can manage and different enterprise solutions often lead to further data confusion [8].

Disparate systems create potential for data error; data errors are these inconsistencies in data that cause data quality issues which could result in lost consumer cross selling opportunities, invoicing problems, or even failed products. It is estimated that incorrect data in retail industry lead to a loss of approximately $40 billion annually [9].

Master Data Management also called Reference Data Management, is an integrated business and IT function that focuses on the management and interlinking of reference or master data that is shared by different systems and used by different groups within an organization [4].

Gartner defines Master Data Management as below.

“Master Data Management is a technology enabled business discipline that helps organisation achieve a “single version of truth” in such important areas as customers, product, accounts etc.

In MDM, the business and IT organisation work together to ensure the uniformity, accuracy, semantic persistence, stewardship and accountability of enterprise’s official, shared master data. Organisation apply MDM to eliminate the costly debates on “whose data is right” which can lead to poor decision making and business performance” [6]

1. Master Data

Master data provides a foundation and a connecting function for business Intelligence(BI) by the way in which it interacts and connects with transactional data from multiple business areas such as sales, service, order management, purchasing, manufacturing, billing, accounts receivable, and accounts payable(AP) [1].

According to master data (also called reference data), is any information that is considered to play a key role in the core operation of a business, typically shared by multiple users and groups across an organization and stored on different systems [4].

Master Data is complementary to BI and can provide an excellent source of dimensional data [11].

2. Data Domains

Master Data consists of information critical to a company's operations. The data is usually categorised master data entity such as customer, products, vendors, partners, employees, inventory etc. These categories are called Data Domains [1 and 9]. The concepts of Master Data Management apply to each of the domains in general. Each of the domains has different implementations challenges.

3. Master Data Management

In an article on Enterprise Data Management (EDM), Cohen [5] describes MDM as one of the main components of an effective enterprise data management (EDM) program. There are six components in enterprise data management (EDM). Figure 1 describes the components of enterprise data management (EDM) which when managed well together help companies to take advantage of the latest technological innovations and more effectively manage their information.

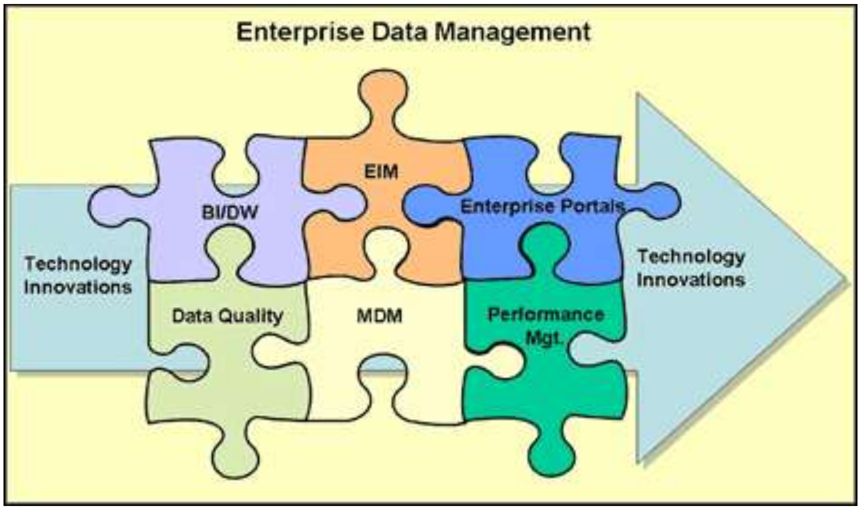

Figure 1: Enterprise Data Management [5]

MDM is the process of helping a company to standardize the definition and attributes of all of its critical data elements (customer, vendor, product, etc.) to create a common point of reference enterprise wide. MDM can facilitate the sharing of data among all a company's disparate business functions, departments and even divisions - not to mention across all information systems, platforms and applications. Without an effective enterprise wide MDM implementation the other components for EDM will not be as effective. The business cases defined in the next section provide some examples to support this statement.

A MDM solution therefore creates a single view of data in any targeted data domain. This is also referred to as the golden record. For example, if the master data management is for Customer Data then any record will refers to the “single truth” or “single customer view” which is an authoritative customer record that has usually been generated by extracting, cleansing the data from multiple channels of enterprise. This process is called Customer Data Integration (CDI). CDI is the subset of MDM and encompasses every aspect of customer touch points in the organisation. CDI is the most widely used implementation of MDM [1] while [8] mention that customer master data is a common starting point for an organization’s MDM.. An effective CDI means that any customer attribute is uniquely identifiable and there exists no multiple versions of customer attribute in any of the company’s enterprise IT systems.

Product data management (PDM) systems are used to manage all product-related data and also product master data. Product master data is far more complex than customer master data [8].

3. Building a Business Case for MDM

In this section four Business Case scenario are provided. These business cases present a problem statement the solution to which is implementation of MDM.

3.1. Business Case #1: Merger of two companies.

In U.S.A, a major telecom service Provider Company A bought telecom service provider company B. The two major telecom service providers merged in 2005. Each of the company individually provided mobile phone services to approximately 20 million subscribers each – one used CDMA technology and other used GSM technology.

The challenges for the newly merged companies where multiple, the chief among them being to consolidate their customer service department such that the outward projection to the customer was one brand. This was in addition to normal integration related problems like, HR, finance and regulatory etc.

The challenges resulting out of this merger that MDM could address are:

a. How do we accomplish consolidation of all customer bases such that there is single source of truth on all the customer attributes? This is CDI part of MDM.

b. Because the two companies had different price plan, devices, and products – they need to be consolidated into one product reference. This is part of PDM or MDM.

3.2. Business Case #2: Replacement of an ERP application

In the year 2008, a major crown corporation, managing social housing portfolio replaced its legacy ERP system with a Commercial off the shelf (COTS) implementation. This had a unintended impact on the downstream applications when the migration to new ERP was completed. The downstream applications that used the original ERP's unique identifier (UiD) to cross reference were now out of sync with the corporate property master data in the corporate ERP because the new application did not use the same Unique Identifier (UiD). Additionally the new ERP did not integrate with the downstream application. That is whenever a attribute is modified in the new ERP, that modification is not communicated to the downstream applications. The new ERP being COTS product, cannot be modified without incurring huge cost.

Thus the challenge here is to ensure to synchronise the data in the down steam applications whenever data is the main ERP is modified. This is a classic case for MDM implementation.

3.3. Business Case #3: New application introduced

In year 2012 a new application was introduced in an organisation that manages building repairs. This application was a web based application which was used to order jobs. The application used the job costing information from the ERP system and communicated back to the ERP system when the order was completed. This required that the job costing information that is actually maintained in the corporate ERP is shared with the new web based application. This is problem can be tackled using MDM.

3.4. Business Case #4: The Homeless Shelter Network

There are many homeless shelters in a big city. Big urban center could have upwards of hundreds of such shelters. All of them are mostly funded either directly by provincial government or a provincial government funding agency and/or individual city councils. However each one of shelter house is a mostly independent not-for-profit organisation or a charity run entity. Each one of the shelters would have their own distinct business processes, and data collection methods with varying degree of sophistication [12].

Each shelter would be able to provide the number of clients it served in a particular time period. However if the funding agency or the governments wants to know how many unique homeless individual were served by the all the shelters funded by it, there is no way of knowing it unless every shelter uniquely and uniformly identify the homeless individual it serves i.e. if each one of them run same software application to manage their shelter or at least use same identity proof. However, in practice not everyone uses same software or same method of indentifying the homeless client.

MDM can be used in scenario like this to overcome data duplication problem.

4. MDM Approaches and Architecture

Before proceeding with MDM architecture it is important to review the types of data and tables in a modern database application. Enterprise systems deal with and generate different types of data. These data are classified into data domains like, customer, products, accounts, vendors etc. Additionally the data can be classified as transactional data and non-transactional data. Transaction data are generally stored in transaction tables. Examples of transaction data include call records of a subscriber (CDRs), or line items in a purchase order or a bank transaction in ATM machine. Normally transaction data tables have large number of records. The data in transaction tables is dynamic and the tables are frequently updated with new rows. The data in the transaction table are generally critical for regulatory reporting. However before the advent of virtual server and cheap storage the transaction data used to be archived in tape drives or sometimes simply deleted after certain time period. The transaction tables provide the point in time information and therefore are at the heart of any Business Intelligence initiative.

Non-transaction data is also called reference data are stored in tables called reference table. The reference table contain such information as customer unique identifier details (name, address, account number etc), vendor details (vendor name, vendor number, vendor address etc), and company employees, company address etc. This information is critical to the organisation. The data in reference tables are used for referential integrity in transaction table. Reference tables are normally never archived or deleted.

Another way of categorising data is operational and non-operational. Operational data is the real-time collection of data in support of a company’s need in their daily activities.

Nonoperational data is normally captured in a data warehouse on a less frequent basis and used of business intelligence (BI) [1].

Accordingly this particular classification of data is used to divide MDM into two sections Operational MDM and Analytical MDM [1, 3, 4, 8 and 13]. A third category is a combination of operational and analytical MDM and is called enterprise MDM [1, 4]. Operational MDM integrate operational applications such as enterprise resource planning (ERP), customer relationship management (CRM), and supply-chain management (SCM) in upstream data flow [8]. Analytic MDM is seen in practices which reminds data warehousing (DW) such as customer data integration and financial performance management. The enterprise MDM system is used for maintain and publishing all the organisation master data.

The architecture of enterprise MDM is shown in figure 2. The main components of MDM system are MDM applications, a master data store, a metadata store and a set of master data integration services [14]. This is shown in figure 3.

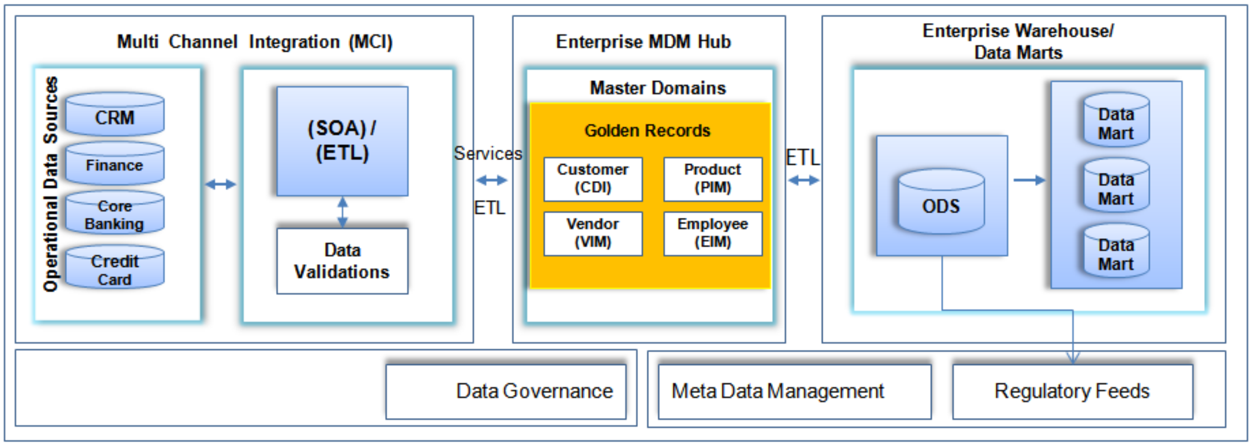

Figure 2: Enterprise MDM Architecture [3]

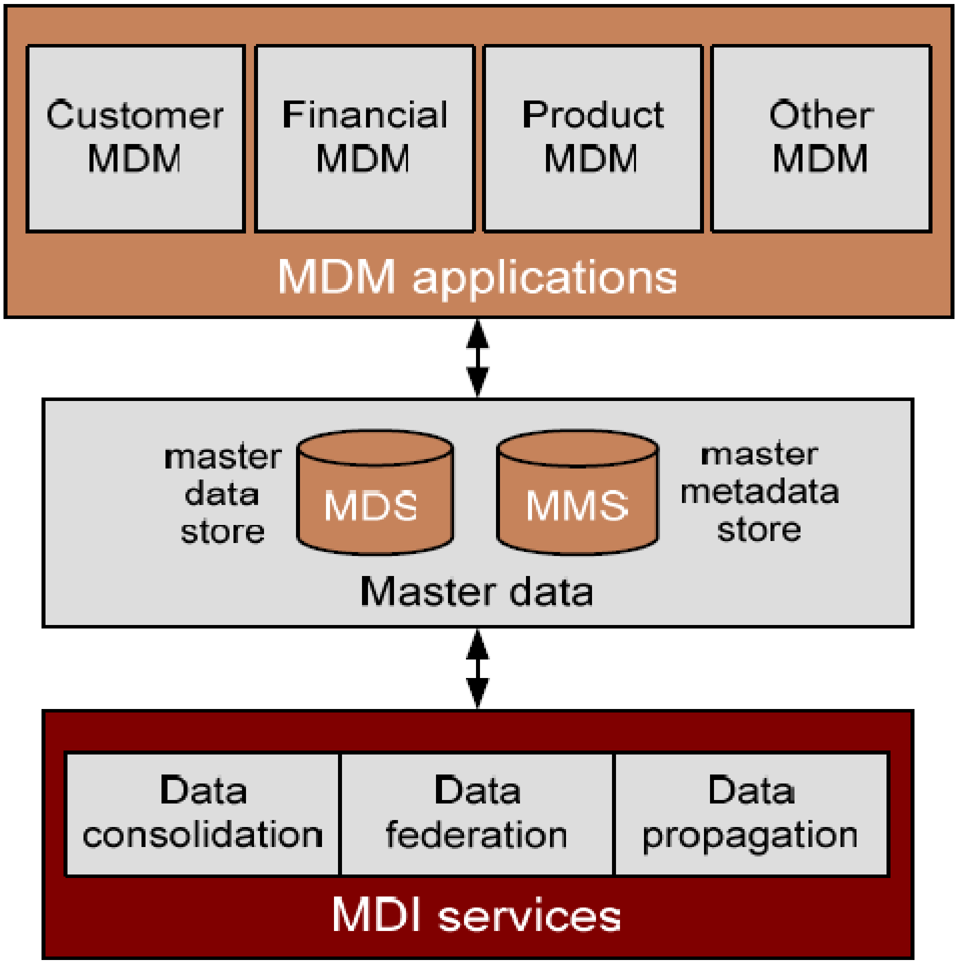

Figure 3: MDM components [14]

Enterprise MDM is the most intrusive implementation, while analytical MDM is least intrusive reason being enterprise MDM encompasses both operational and non-operation data. As a result the gain is highest in enterprise MDM implementation. Additionally while implementing MDM, it makes sense to break down the MDM initiative into phases and target just a few applications at a time to avoid disruption.

5. MDM Framework

It is important to understand how master data is created, used, maintained and integrated with multiple applications. The MDM frameworks mentioned in this section describes various ways to store, process and synchronise master data. The main components of MDM are:

1. Composite applications

   The applications are the IT applications which will collect, use and maintain master data. An example of this application could be customer service software used in a call-center, an ERP system, a downstream application, a front end web application etc. Each composite application will have its own database or two applications could share a database.

2. Business Process Orchestration

   This is the most critical part of MDM initiative. Business Process Orchestration is a set of rules, guidelines, workflow or regulations created by the business owners and leaders such that the data being entered in the applications are consistent, and accurate. An example of this role could be as below. To eliminate discrepancies in name (Michael vs. Mike, Robert vs. Robert) of the same person, a homeless shelter clerk would verify the name of the client with his MCP card or any government issued valid ID. This is a an example of a simple business process and correct implementation of this is critical for success of MDM. A complex example of business process could be a set of Microsoft SharePoint workflow steps that would be required by a clerk to be completed before an new vendor or a supplier is added to the SCP application.

3. Enterprise Service Bus

   This is the technology component of the MDM. This could be a complex middleware products like Software AG's Webmethods, IBM websphere or Tuxedo. Or it could be a simple solution as a network share with xml reader products.

4. MDM data synchronisation services

   These are services that will synchronise master data between the applications or between application and the master data store. These could be triggered based service or could be message based service.

5.1. Single Central Repository Architecture (SCRA)

The Single Central Repository Architecture is shown in figure 4. In this architecture, the master data is stored in a single central repository which will be updated by the MDM services, and the applications. The applications will not hold a copy of any master data. Applications refer to master data from the central repository. There are no local versions of Master Data anywhere.

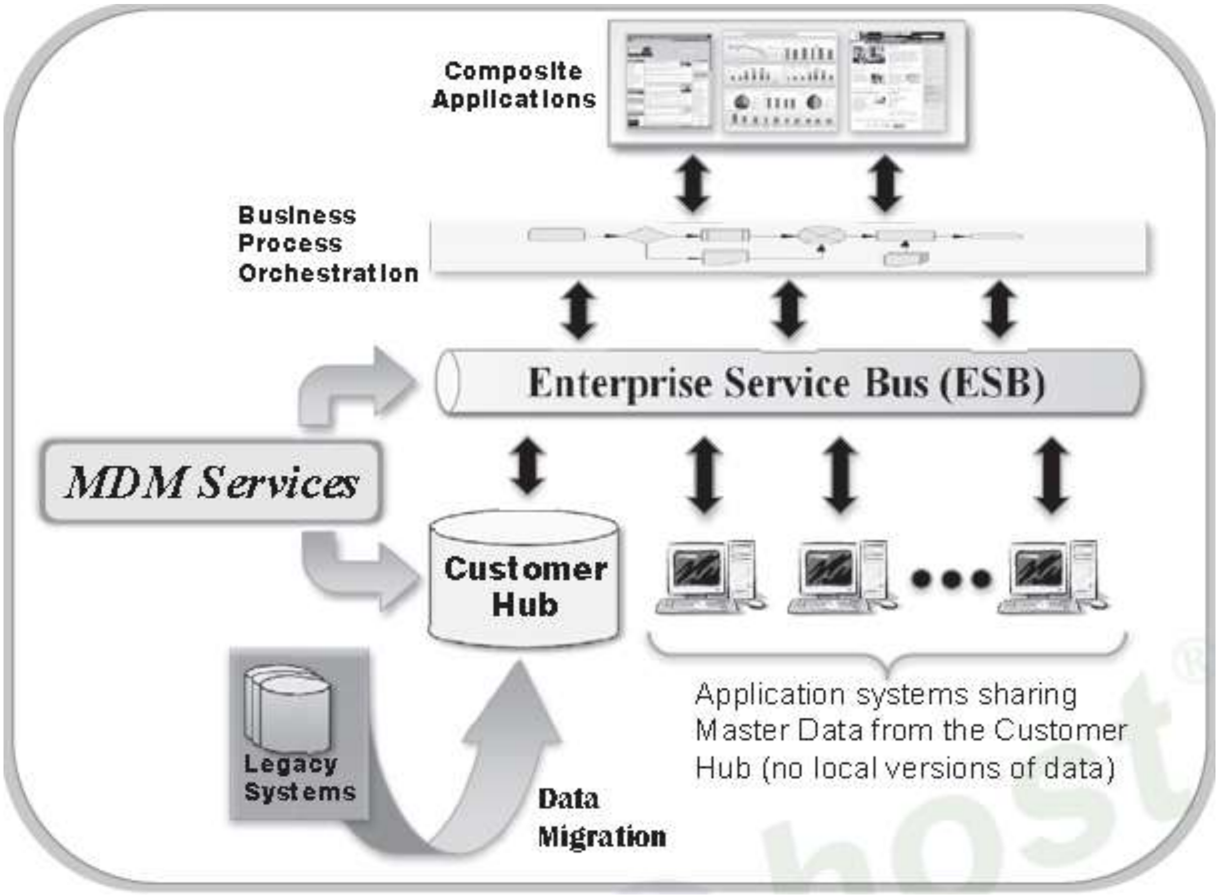
Composite Applications
Business Process Orchestration
Enterprise Service Bus (ESB)
MDM Services
Customer Hub
Legacy Systems
Data Migration
Application systems sharing Master Data from the Customer Hub (no local versions of data)

Figure 4: Single Central Repository Architecture (SCRA) [1]

Advantages of SCRA are that it guarantees data consistency [1], and some of the applications may become redundant one SCRA repository is up and running therefore enabling to retire legacy applications.

However the disadvantage is the massive upfront cost. The upfront implementation and migration to SCRA is costly because it requires massive data conversion effort and migration of data from multiple disparate systems. This can also be disruptive to business.

The prevalence of COTS products could possibly make implementation of SCRA difficult if not impossible. However once SCRA is implemented, the cost of maintenance would be minimal [1].

5.2. Central Hub and Spoke Architecture (CHSA)

The central Hub and Spoke Architecture is a variation of SCRA [1]. It contains a central repository (central hub) while also providing ability to the individual application to maintain an extension of the data. Therefore some application would access master data from the central hub and not keep a local copy, others might only use the central hub as a reference [15].

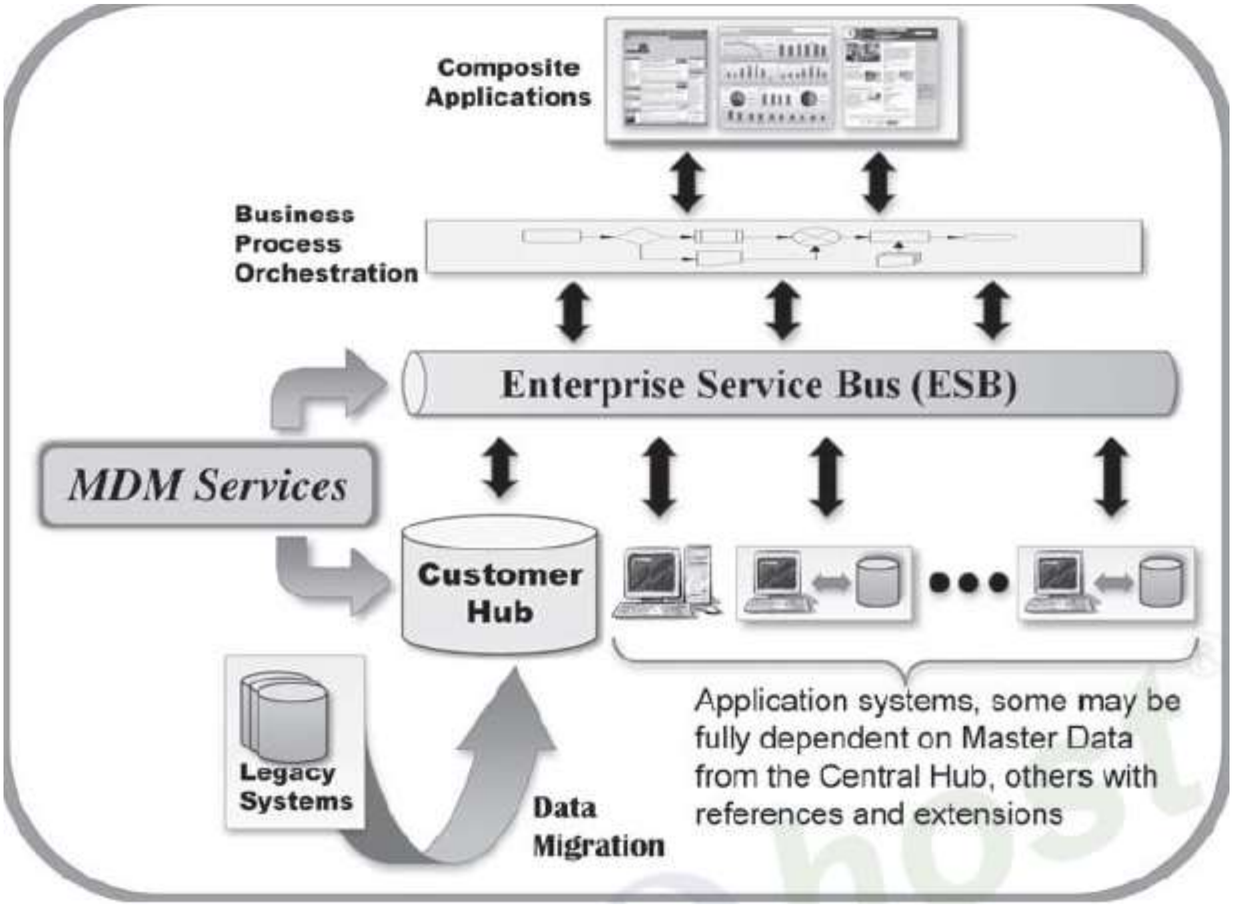
Composite Applications
Business Process Orchestration
Enterprise Service Bus (ESB)
MDM Services
Customer Hub
Legacy Systems
Data Migration
Application systems, some may be fully dependent on Master Data from the Central Hub, others with references and extensions

Figure 5: Central Hub and Spoke Architecture (CHSA) [1]

The biggest advantage of CHSA is its flexibility to relatively decouple by supporting spoke systems. This flexibility is really important when we have COTS applications which cannot be coupled with the central hub [1]. The flaw of CHSA hub-and-spoke is that it doesn't address issues of timeliness and latency [15]. Additionally the data conversion effort is still required.

5.3. Virtual Integration (VI)

This pattern uses data virtualization to provide one or more on-demand integrated views of master data entities such as customer, product, asset, employee etc. even though the master data is fractured across multiple underlying systems. Applications, processes, portals, reporting tools and data integration workflows needing master data can acquire it on-demand via a web service interface or via a query interface such as SQL [16].

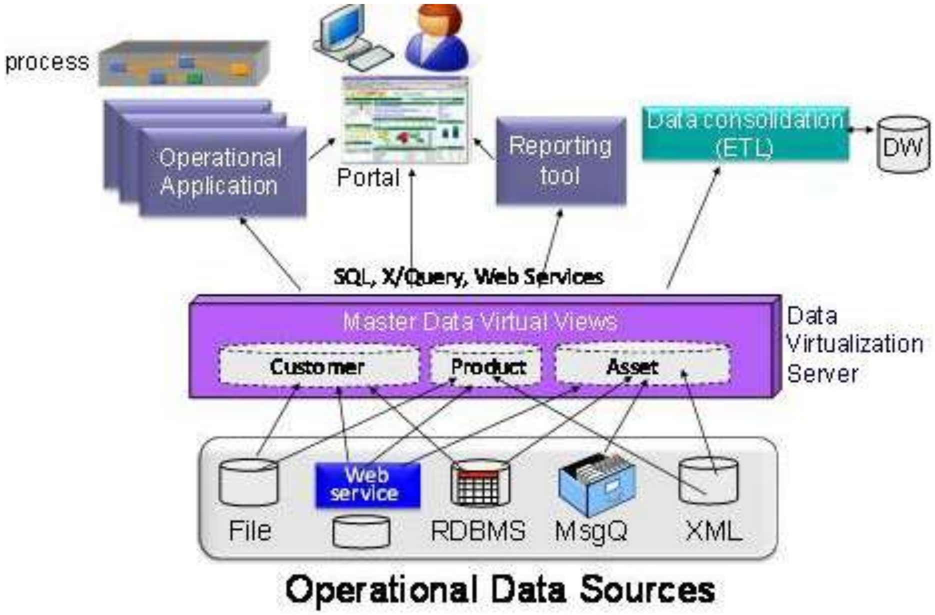

Figure 6: The Virtual Master Data Management pattern

5.3.1. Data Service Federation (DSF)

Data Service Federation is a common Virtual Integration architecture. The virtual integration pattern aggregates data from multiple sources into a single view by maintaining metadata definition for all the sources [1].

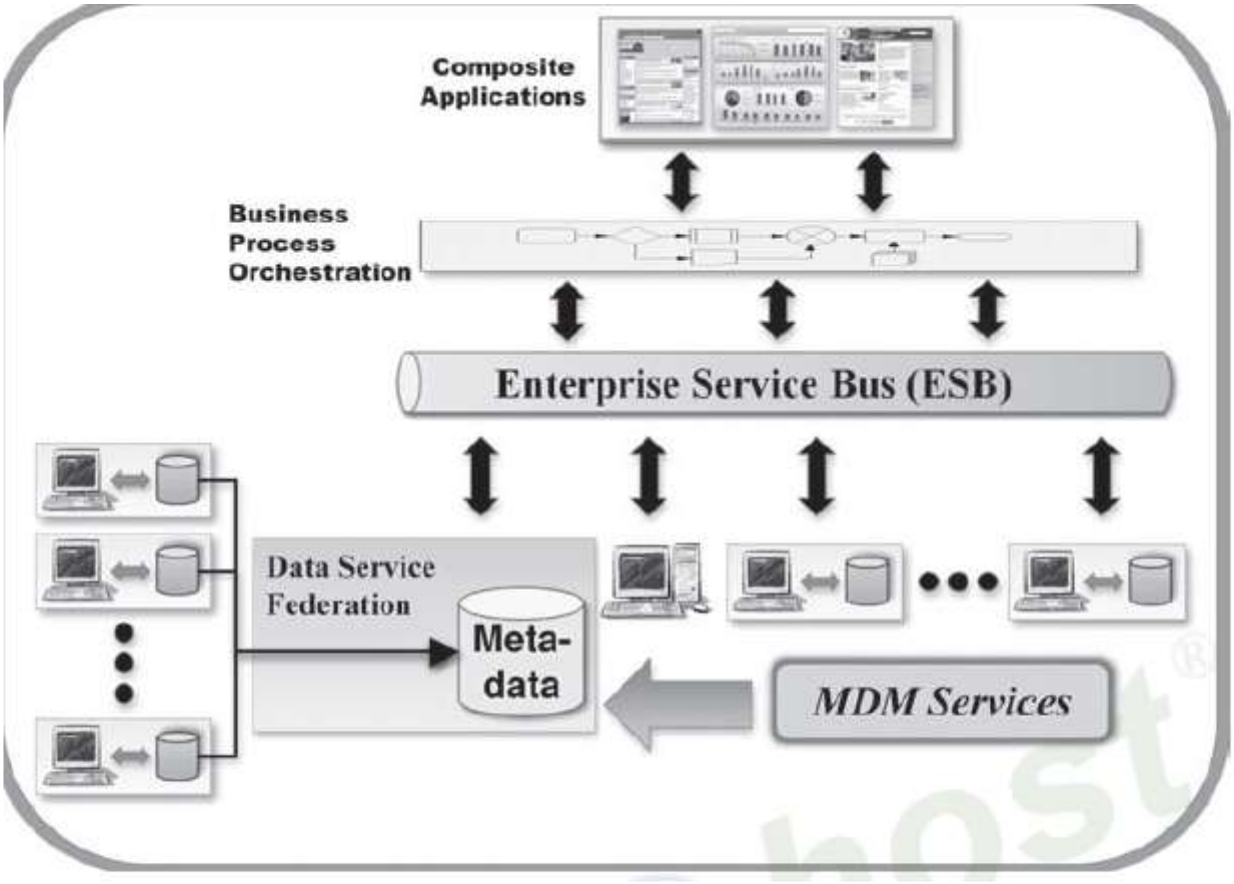

Figure 6: Data Service Federation (DSF) [1]

The advantages of DSF is less costly than the SCRA and CHSA because the data does not have to be physically copied from one location to another nor any additional storage space is required. However the biggest disadvantage is that the data improvements are not propagated back to the source application [1].

6. Data Synchronisation Techniques

The data synchronisation is the critical step to maintain the consistency of master data. Synchronization step is mostly required regardless of what type of MDM framework is implemented. In this paper three different types of data synchronisation techniques are described. However it must be noted that there can multiple other ways of synchronisation data and that database or data synchronisation is not unique to MDM.

6.1. Trigger based

Trigger based approach is described in the figure below. In this approach the trigger for data synchronisation is an update or insert event on the source database table record.

In this step when a candidate record is modified in source database, a service polls the event and propagates the modified data to other database tables.

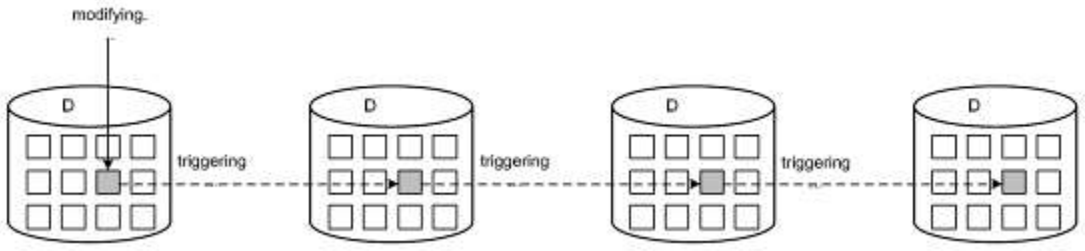

Figure 8: Trigger based [13]

This kind of synchronisation ensures that all the data in multiple tables are always synchronised. However while it is easy to implement in a small scale setting, the process is extremely computation intensive in a large scale. It is also dependent on high availability of networks.

6.2. Message-based Data Synchronisation and Integration Framework (MDSIF)

The message based data synchronisation and Integration Framework is detailed in the article [13]. In this process message oriented middleware (MOM) like IBM's MQ Series is used to propagate the data from multiple data

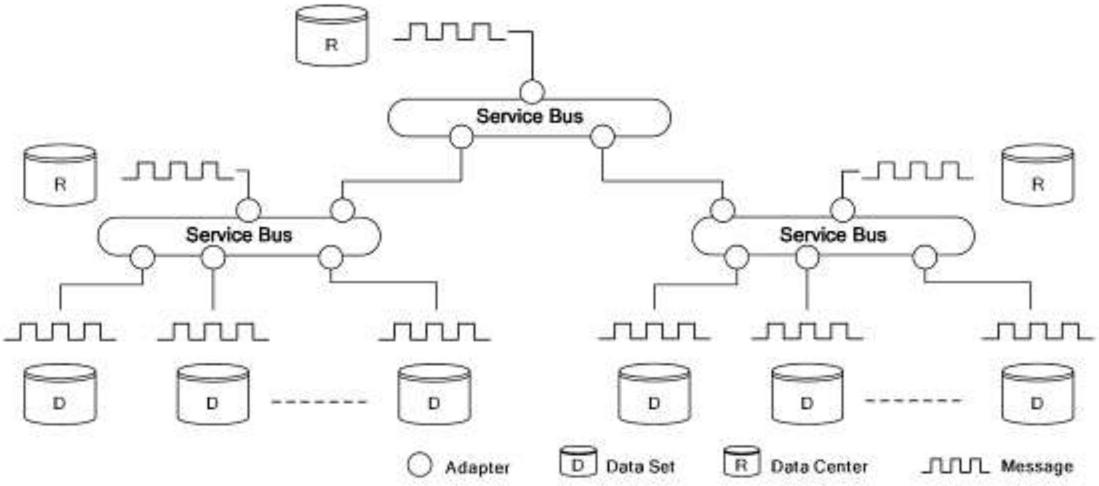

Figure 7: Message-based Data Synchronisation and Integration Framework (MDSIF) [13]

MDSIF bring all the message based middleware's advantage that is has an advantage in that it does not have to depend on network availability. Additionally it does not add load to the network thus avoiding performance bottleneck [13]. However these are expensive piece of software and require lots of investments in infrastructure and high maintenance cost.

6.3. Conflict resolution

6.3.1. Confidence Tables Approach

When multiple data sources are involved there will be data conflicts. These conflicts are different from database conflicts. Confidence table approach to resolve data conflicts mentioned in [13] and is effective tool to manage data conflicts.

In this approach each data that is under consideration are given a confidence level based on the source that has modified the data. This confidence is designated in the business layer processes. When data synchronisation process finds data conflicts it refers to the confidence table. The source that has higher confidence level will get higher preference and the data with lower confidence level is rejected.

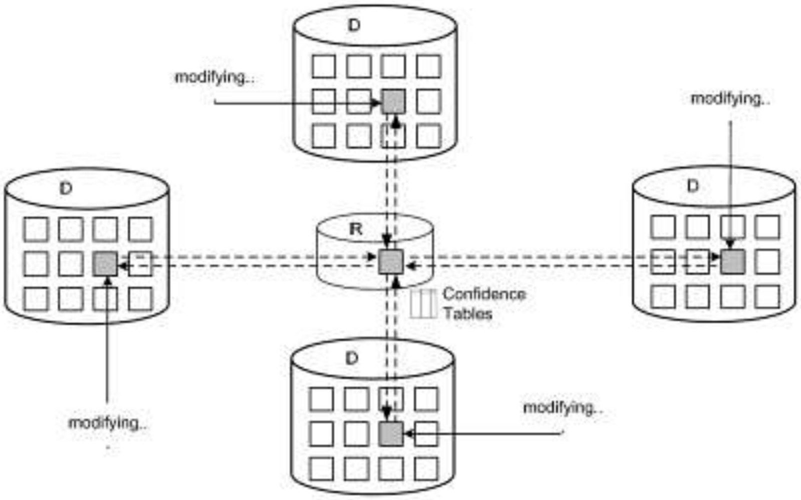

Figure 9: Confidence Tables Approach [13]

The method of resolving conflicts during synchronisation is dynamic and is easy to maintain. However it needs upfront identification and constant maintenance. Additionally input from business owners is critical in this approach.

7. Case Study

In this section we will revisit a business case mentioned in the previous section and apply MDM principles to achieve one Master Data. We will select case study on multiple homeless shelters mentioned in section 3.4. The synopsis of the problem statement for this case is mentioned below.

Problem statement

There are multiple homeless shelters that are funded by provincial government agencies. Each of the shelters is independent entities which have their own business processes, procedures and, software.

This situation implies that when a homeless client walks into a shelter, because there is no uniform business processes and because the shelters do not share database following issues exist:

1. A client may not be identified uniformly by all the shelters. That is shelter A might register a person with only first name, while shelter B might register the same person with last name and initials.
2. Suppose the funding agency wants to know how many homeless were served by the shelters, there does not exist a single place which will hold this information. Agency cannot track the fund usage, efficiency of usage and perform trend analysis.

Proposed MDM solution

Given the background of the problem statement, it is proposed that we will use Data Service Federation architecture with message based data synchronisation. Additionally we will use confidence table approach to resolve any conflicts.

Firstly business processes are implemented to ensure that each shelter records the name correctly. The solution could be as simple as using same identification card to correctly identify first name and last name and other identification information. Additionally a workflow could be implemented such that when a new client is registered a review process happens and the record is approved by a manager. This step will eliminate any clerical error during data entry.

As discussed in previous section virtual integration framework uses web services and metadata definition to uniquely record and integration MDM data between composite applications. In our case the composite applications are the applications in each of the shelter. Thus a MDM master repository is created in a centralised location and the view made available to other composite applications.

To ensure data synchronisation with the master data repository we will implement message based data synchronisation technique. This will eliminate network related issues that can impede a trigger based approach.

Confidence tables are created to resolve data conflicts. In the confidence table will hold the confidence level for each information record that is required to be tracked by the funding agency. When a field is updated by multiple applications then the source with the highest is given priority and that information is used to update in the repository.

8. Conclusion

It can be concluded based on the findings in this paper that MDM cannot be classified as only an IT problem but it is a managerial challenge which requires structural changes to managing business processes, and managerial decision making. Without engagement of the business owners MDM cannot succeed.

In the case study we found that the funding agency can now accurately track the number clients that were served and also provide a bigger picture for policy makers on the number of clients using the shelters.

Thus MDM improves overall data quality and confidence. With successful implementation of MDM, a single source of truth exists regarding any master data that is tracked – customer, products, suppliers etc. This will help businesses make proper business decisions with the confidence that the data on which these decisions are based is accurate.

9. References

1. Master Data Management in Practice: Achieving True Customer MDM. Cervo, Dalton, Allen, Mark. ISBNs: 9780470910559. 9781118085660. [Wiley Corporate F&A].Hoboken, N.J.: Wiley. 2011
2. Management of the master data lifecycle: a framework for analysis. Ofner, Straub, Otto and Oesterle.
3. Practical Approach for Master Data Management, Chandra Sekhar Bhagi, World of Computer Science and Information Technology Journal (WCSIT) ISSN: 2221-0741 Vol. 1, No. 5, 213-216, 2011
4. Enterprise Master Data Management Trends and Solutions. APOSTOL, Constantin-Gelu, http://revistaie.ase.ro, Vol. XI, no. 3/2007
5. Understanding the Scope of Data Management: The Components of a Robust Enterprise Program. Cohen, Rich. http://www.information-management.com
6. Gartner Inc,."Hyper Cycle for Master Data Management, 2010" Andrew White and John Radcliff. https://www.gartner.com/doc/1464920/hype-cycle-master-data-management
7. The transverse information system: new solutions for IS and business performance, Rivard, François. 1-84821-108-2, 978-1-84821-108-7 Date: 2009 Page: 49 – 84
8. Managing one master data – challenges and preconditions, Risto Silvola, Olli Jaaskelainen, Hanna Kropsu-Vehkapera, Harri Haapasalo. Industrial Management & Data Systems, ISSN: 0263-5577, Volume 111 issue 1

9. Methodologies for data quality assessment and improvement, Batini, C., Cappiello, C., Francalanci, C., Maurino, A. , *ACM Computing Surveys*, Vol. 41 No.3
10. How Kijiji's data threw off Ottawa’s math on skills shortages. http://www.theglobeandmail.com/news/politics/how-kijijis-wonky-data-threw-off-ottawas-math-on-skills-shortages/article17675622/
11. Introduction to Master Data Management. Mark Rittman, www.rittmanmead.com
12. National Homelessness Information System http://hifis.hrsdc.gc.ca/initiative/index-eng.shtml
13. Message-Based Approach to Master Data Synchronization among Autonomous Information Systems, Dongjin Yu and Hangzhou Dianzi. International Journal of Enterprise Information Systems, 6(3), 33-47, July-September 2010.
14. Using Master Data in Business Intelligence Colin White, BI Research available at www.fm.sap.com and www.broadstreetdata.com
15. The Flaw of the Hub-and-Spoke Architecture, Evan Levy, Information Management Jounal available at http://www.information-management.com/blogs/business_intelligence_bi_hub_and_spoke_architecture_analytics10015083-1.html
16. Data Federation- Master Data Patterns - The Virtual MDM Pattern, Mike Ferguson, available at http://www.b-eye-

network.co.uk/blogs/ferguson/archives/2009/12/data_federation-_master_data_p.php