

8-2010

An ERP-centric Master Data Management Approach

Alexander Maedche

University of Mannheim, maedche@eris.uni-mannheim.de

Follow this and additional works at: <http://aisel.aisnet.org/amcis2010>

Recommended Citation

Maedche, Alexander, "An ERP-centric Master Data Management Approach" (2010). *AMCIS 2010 Proceedings*. 384.
<http://aisel.aisnet.org/amcis2010/384>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2010 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

An ERP-centric Master Data Management Approach

Alexander Maedche

Chair of Information Systems IV - Enterprise Information Systems (ERIS)

University of Mannheim

<http://eris.bwl.uni-mannheim.de>, maedche@eris.uni-mannheim.de

ABSTRACT

Both, ERP and MDM target integration: ERP follows a business process centric approach and integrates master data along value chains into one ERP system. MDM follows a broader approach and targets providing an enterprise-wide single version of the truth of selected master data objects and distribution of them to the different target IT systems. ERP and MDM include overlapping master data management functionality and the question appears how they can complement each other specifically in the case of complex ERP landscapes. This paper describes the results of a multiple-case study analysis and interpretation of MDM approaches in the context of ERP landscapes. We identify and discuss major business use cases, a conceptual architecture and requirements for an ERP-centric MDM approach in this paper.

Keywords

Enterprise Resource Planning, Master Data Management.

INTRODUCTION

Enterprise Resource Planning (ERP) represents an enterprise view of the business: a view of a company and all its parts as connected whole, rather than small silos of activity. ERP describes chains of activities of a company, also known as value chains. The value chain concept was developed and popularized by (Porter, 1985). One major goal of ERP systems is to integrate all the relevant master and transactional data of the enterprise and let people and processes operate on the same data objects (Monk and Wagner, 2006). Several authors have confirmed that ERP projects always include master data consolidation and harmonization activities as part of the overall introduction activity. (Vosburg and Kumar, 2001) elaborate the importance of shared understanding of master data within the context of the business. (Welch and Kordys, 2007) emphasize master data management as one of the major challenge of ERP projects. Furthermore, over the time, the number of installed ERP systems in large enterprises has been growing, extended enterprise systems such as PLM, SCM or CRM have been introduced and recently Software-as-a-Service-based extensions such as talent management applications in HR have been used. Enterprise Resource Planning achieved a lot in the last 15 years regarding standardization of master data objects, but definitely didn't solve the overall problem. Actually, with the rise of complex ERP landscapes in large enterprises it created a new master data management challenge.

Master data management (MDM) is also pursuing consolidation of data into a single version of the truth. However, in contrast to an application-/process-centric approach like ERP, MDM follows an application-agnostic approach and tries to define and maintain consistent definitions of master data and enable sharing across multiple IT systems within an enterprise. (Dreibelbis, A., Hechler, E., Milman I., Oberhofer, M., van Run, P. and Wolfson, D., 2008) distinguish three key methods of use of MDM: First, collaborative authoring MDM supports the definition, creation, and synchronization of master data. It is associated with the creation, augmenting, or altering of master data to support processes, such as the new product introduction and definition process, product catalog management or data stewardship. Collaborative authoring MDM requires services to support workflow and check-in, check-out services to control the creation, management, and quality of master data. Second, operational MDM supports the consumption of master data by operational systems to perform

transactions and the MDM repository is considered the authoritative source of master data. Master data is leveraged by applications through services, where services provide control over master data creation, management, quality, and access. In operational MDM, typically only a subset of the master data is stored centrally and the different operational systems keep control over the detailed master data objects. Third, analytical MDM focuses on providing an accurate, clean source for master data to provide the dimensional source for analytical environments, specifically Data Warehouses or more general Business Intelligence platforms.

The four different implementation styles i) consolidation, ii) registry, iii) coexistence and iv) transactional hub are distinguished by (Dreibelbis et al., 2008). Because of the growing complexity from i) to iv), one has often been introduced MDM in enterprises in the form of a dedicated standalone solution; a master data approach that is only loosely coupled with existing operational enterprise systems. This approach can work fine for analytical or collaborative MDM methods of use, including functions like data entity consolidation and harmonization, data consistency, data cleansing, enrichment & de-duplication. However, the loose coupling approach often results in issues in operational MDM scenarios where multiple ERP systems and other enterprise systems are involved. Specifically, one is faced with challenges regarding data replication and structural and semantic mapping of objects between MDM and ERP.

This paper describes the outcome of a multiple-case study research analyzing operational MDM usage in the context of ERP system landscapes in large enterprises. The paper is structured as following: First, we start with an overview on the most relevant MDM use cases that we have identified in the context of enterprise systems literature, specifically ERP systems usage in large enterprises. We describe the setup and the results of our case study research approach and derive from there two fundamental business use cases of MDM in ERP. Second, we define a conceptual architecture and requirements for an ERP-centric MDM approach. Third, provide an overview on related work and put this paper in the context of it. Before we conclude, we provide a summary and an outlook on future work.

RELATED WORK

Integration from a technical perspective can take place at different abstraction layers. In general one can roughly distinguish between presentation integration, functional integration and data integration. Presentation integration targets user interface level integration. Enterprise Portals represent sophisticated presentation layer integration approaches. Functional integration describes integration at the business logic layer. Enterprise application integration (EAI) approaches based on middleware platforms follow a functional integration approach. Recently, the concept of enterprise services buses (ESB) in the context of service-oriented architectures (SOA) has evolved. Finally, integration on the data layer describes ways to harmonize data contained in different systems (Halevy, Rajaraman, Ordille, 2006). From a technical perspective, integration has been intensively research and documented, e.g. in the form of integration patterns (Hohpe and Woolf, 2003) and comprehensive IT integration approaches (Britton and Bye, 2004). (Hasselbring, 2000) has looked at integration from an information system perspective and came up with the three layers of business architecture, application architecture and technology architecture.

The principle concept of master data management has been mainly approach from a practical perspective (see for example the books of (Loshin, 2009), (Berson and Dubov, 2007) and (Dreibelbis et al., 2008)). Especially (Dreibelbis et al., 2008) looked into enterprise system integration aspects in more detail. In their excellent book they describe for example a dedicated SAP application integration blueprint and elaborate on the different abstraction integration layers as mentioned above. In contrast to our work, they look at the integration challenge from a more technology driven perspective and less consider the actual business scenario. (Loser, Legner, Gizanis, 2004) look at specific challenges of managing master data in service processes, present conceptual approaches for the distribution of master data taking into account distributed and heterogeneous applications and illustrate how customer and product master data management supports innovative service processes based on the case study.

Not much work has been done in the context of looking at integration and specifically master data management from a business perspective, specifically in the context of enterprise systems. Several authors have confirmed that ERP projects always include master data consolidation and harmonization activities as part of the overall introduction activity. (Vosburg and Kumar, 2001) elaborate the importance of shared understanding of master data within the context of the business. (Welch and Kordys, 2007) emphasize master data management as one of the major challenge of ERP projects. (Dayton, 2007) has described typical MDM challenges when using a combination of ERP and CRM in the context of holistic performance management approach.

(Knolmayer and Roethlin, 2006) have analyzed MDM approaches in the context of ERP landscapes. They carried out a case study on distributed ERP systems in a large corporate group and its 13 divisions, focusing on data quality issues in logistics. The authors conducted more than a dozen interviews with representatives from all major divisions, covering different roles

including senior managers, controllers, material managers, process owners, IT specialists, and public relations managers. The key legacy applications had been migrated and consolidated into ten SAP R/3 system instances. However, divisions were allowed to implement, adapt, and extend the individual R/3 systems to their individual business needs and to handle implementation projects with their own resources, consultancies, and methodologies. A dedicated master data hub based on SAP R/3 has been installed; the different R/3 instances exchange data via SAP's ALE concept (application linking and enabling). The hub provides a "one-to-many" distribution of master data values to the divisional ERP instances, where they are completed and adapted to divisional needs. The authors describe several problems that come with the described landscape, specifically lack of consolidation of data records, data warehouse quality issues when consolidating data, lack of global view for material requirements planning process. Furthermore, they give ideas on improving data quality along the three dimensions: people, process and technology. They consider reworking an existing base of master data as pretty difficult in an ERP context and organizational barriers cannot be removed by a simple adoption of technology.

The need for managing master data within ERP landscapes has been approached so far mainly from a practical and pragmatic perspective. SAP, for example, has established the ALE concept¹ already quite some time ago. Using ALE, SAP and non-SAP system can be integrated by using synchronous and asynchronous communication without implementing a central integration hub. ALE acts as a layer that incorporates the inter-linking of business objects at the application level following a point-to-point approach. The ALE layer provides application, distribution and communication services as part of each SAP ERP instance. Beside the ALE interface, SAP later developed the SAP PI integration platform which allows transformation, monitoring, validation, etc. Both, ALE and PI do not focus on master data management specifically, they mainly focus on replication, distribution and transformation. For MDM scenarios, SAP offers the SAP Netweaver Master Data Management solution.

Oracle provides a suite of master data management solutions targeting different application scenario with dedicated hubs, such as Customer Hub, Supplier Hub or Product Hub. For the hubs, Oracle offers so-called process integration packs targeting out-of-the-box integration with Oracle business applications such as Siebel CRM or the E-Business Suite. Oracle talks in one of their white papers² about "MDM aware" applications, which are application that supports its original functions, yet behave such that an external system can hold its master information.

Finally, a lot of practical and conceptual work in the master data management area has been done at IBM (Dreibelbis et al., 2008). IBM provides a comprehensive MDM platform called Infosphere addressing the different usage scenarios of MDM as described earlier. The Infosphere Datastage pack provides out-of-the-box connectivity with most important enterprise system solutions such SAP ERP and the Oracle Suite including Siebel, JD Edwards and Peoplesoft.

RESEARCH APPROACH

To understand MDM usage scenarios in the context of ERP systems from a business perspective better, we have followed the case study research approach as described by (Yin, 2009). A case study is an empirical inquiry that investigates a contemporary phenomenon in depth and within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident. According to (Yin, 2009), we went systematically through the iterative phases of planning, designing, preparing, collecting, analyzing and sharing as depicted in Figure 1. In the planning phase, the research questions and rationales for doing a case study approach are identified. The decision for the case study method is made on transparent facts, understanding strengths and limitations of this research method. In the design phase, the unit of analysis and the potential cases are defined. Existing published work is analyzed; theories, propositions and issues are developed. Based on this the actual case study design is decided (e.g.

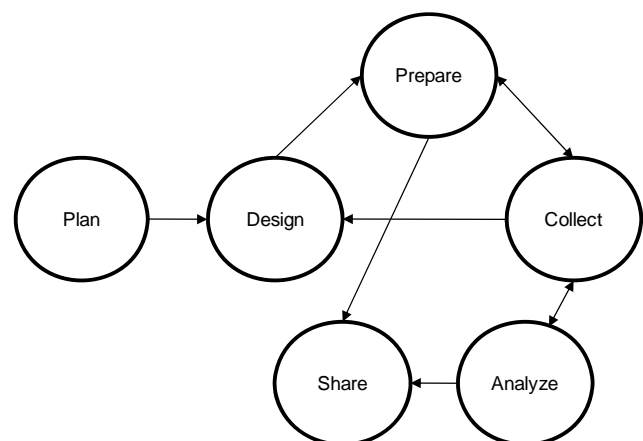


Figure 1. Case Study Research Approach (Yin, 2009)

¹ <http://www.sdn.sap.com/irj/sdn/ale>

² <http://www.oracle.com/applications/plm/pip-for-oracle-product-hub-data-sheet.pdf>

single vs. multiple). In the preparation phase, the boundary conditions for executing the case study are set. Specifically, case study investigators are prepared and trained and a protocol structure is developed. Ideally, a pilot case is conducted to check feasibility. Within the collect phase, the actual case study is carried out following the defined protocol. (Yin, 2009) describes six sources evidences (e.g. Interviews, direct observations, physical artifacts ...) to be investigated as part of this phase. The collected data should be captured in a case study database. Furthermore, the chain of evidence should be captured. In the analyze phase, one may follow four general analytical strategies: 1) relying on theoretical propositions, 2) developing a case description, 3) using qualitative and quantitative data and 4) examining rival explanations. Specifically to the analysis of multiple cases, (Yin, 2009) propose a fifth technique called cross-case synthesis. Finally, in the share phase the case study results are reported.

CASE STUDY RESEARCH

For our case study “operational MDM usage scenarios in ERP environments” we followed the case study research approach proposed by (Yin, 2009) as introduced above. In the following, we walk through the six individual phases and describe our major decisions and resulting activities within these phases and selected results per phase.

Plan

In the plan phase the reasons for following a case study approach need to be analyzed and explicitly articulated. Furthermore, one needs to understand the strengths and limitations of the case study research approach. We selected the case study research approach because we are asking “how” or “why” explanatory questions, do not have control over behavioral events and focus on a contemporary phenomena within a real-life enterprise context. Our questions require an extensive and in-depth description of the phenomenon of master data management challenges of complex ERP environment in large enterprises.

Design

The research design in case study research follows typical empirical research patterns. We first identified and described the central research questions. As mentioned above, we deal here with exploratory “why” and “how” questions. Relevant research questions for the work described in this paper are the following:

1. How do large enterprises manage operational master data in complex enterprise system landscapes today, how is distribution and replication between different ERP systems realized?
2. Why are enterprises faced with operational master data management challenges in complex ERP landscapes?
3. How can ERP and MDM complement each other in the case of complex landscapes?

Second, to focus further the scope of the study, it is recommended by (Yin, 2009) to articulate propositions. Our study has many exploratory aspects, so it is not trivial to come up with a good set of propositions. However, we could identify one key proposition for our study: Enterprises that have installed many ERP systems do have an operational master data management challenge. Third, the unit of analysis describes what the case is. In our situation, the case is management of master data within complex enterprise system landscapes in large enterprises. We look at organizations and analyze their currently implemented enterprise information management approach for enterprise systems, with specific focus on master data management.

Finally, we decided to carry out a multiple-case study, because we wanted to get a broader picture of cases selected from multiple large enterprises to be able to produce a stronger effect regarding identification of valid generic patterns and analytical conclusions. We decided to follow a replication approach by selecting cases with similar boundary conditions. Furthermore, we used holistic design by looking at the same unit of analysis within each individual case.

Prepare

Our case study protocol, included field procedures, case study questions and a guide for case study report. Overall, three case study investigators have been involved. We decided to approach IT representatives at corporate IT departments being responsible for the ERP landscapes. All companies participating in our multiple-case study were internationally operating, large enterprises. The companies represented a cross-industry coverage, including the automotive industry, the process industry and retail/wholesale industry. We did one pilot case study with a division of one large enterprise company to pre-evaluate our defined protocol.

Collect

We approached 8 large enterprises and specifically IT representatives at corporate IT departments for our study. For collecting data we primarily carried out focused interviews with responsible IT representatives. If available, we additionally collected copies of documents describing actual ERP landscapes and the data flows between the different systems.

Analyze

According to (Yin, 2009), techniques for analyzing case study results are still not well developed. Unlike well established statistical analysis methods, there are few fixed formulas or even cookbook recipes. Key element in analysis of cases is identifying patterns.

Let us first look at some descriptive data on the enterprises and associated cases we have analyzed. The number of ERP systems installed and used in these companies was in between 3 and 50. Beside ERP systems, legacy systems and extended enterprise systems such as CRM were typically installed and used. The most relevant business objects that are exchange between the systems are product, customer, accounting data and suppliers. In 4 out of the 8 cases a very pragmatic and ERP-only focused MDM approach has been carried out. In these cases either one of the existing ERP systems has been defined as the leading MDM hub or a dedicated MDM ERP system for master data creation and maintenance has been established. One of our interviewee emphasized specifically *“the importance of deep integration specifically on the semantic level of business objects”*, which he considered as the major reason for following an ERP-only MDM approach. Master data between these individual ERP systems is distributed by point-to-point connections or a dedicated EAI platform. In the remaining 4 cases dedicated MDM hubs for business objects within specific scenarios (e.g. product catalogs) have been installed and used. In general, all cases are characterized by hybrid MDM approaches with a different degree of centralization.

Based on a cross-case analysis, we could identify two major business use cases for master data management in the context of ERP system landscapes: 1) Collaboration in Corporate Groups and 2) Heterogeneous System Landscapes. The two business use cases are visualized below.

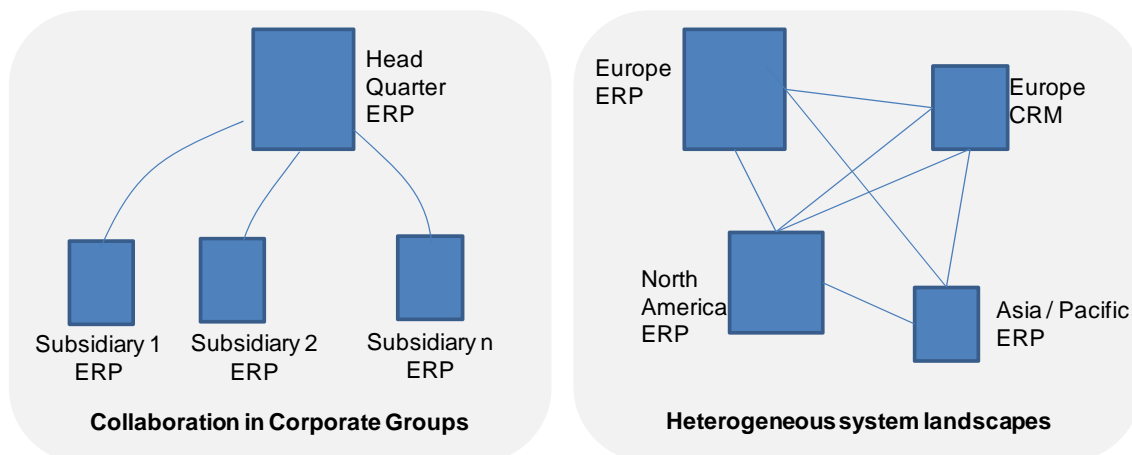


Figure 2. Business Use Cases of MDM in ERP environments

In the following, we shortly describe these two major large enterprise business use cases of master data management in the context of ERP system landscapes.

Business Use case: Collaboration in Corporate Groups

This use case is characterized by a star topology with a head quarter in the center and subsidiaries outside. Typically in such setups there are multiple legal entities involved. Specific central processes run in the headquarter ERP system (e.g. payroll), other processes run locally in subsidiary ERP systems. One can consider the following example taken from the human resources domain: In the head quarter of an enterprise, the organizational structure and employee related master data are maintained. In subsidiaries, processes like leave request approval, browsing in the company address book, project planning etc. are executed locally (independently from the head quarter). Changes of relevant master data in the local systems are organizationally prohibited, they need to be done centrally and received from the master system. This use case patterns has been also identified by (Knolmayer and Roethlin, 2006) in their case study.

Business Use case: Heterogeneous system landscapes

This use case is characterized by a mesh topology with direct peer to peer communication, without a clear master. There are no central processes, all processes run locally. The level of heterogeneity depends on the number of different systems involved. In the easiest case, there are only ERP systems from the same software supplier, most probably on different release levels. In reality, one usually is faced with the fact that beside ERP systems, there is also other enterprise systems involved, including CRM systems, ERP systems from other suppliers, SCM systems or SaaS-based extensions like talent management. One can consider the following example taken from the finance domain: In a consolidation scenario, data from different local systems is consolidated centrally. Data in local systems have to contain information about global entities. Local entities have to be mapped to global entities (e.g. local chart of accounts to global chart of accounts). This use case has been also described by (Dreibelbis et al., 2008), they distinguish specifically between a harmonization use case and the “MDM as the golden middle” use case.

CONCEPTUAL ARCHITECTURE AND REQUIREMENTS FOR AN ERP-CENTRIC MDM APPROACH

The challenge of implementing an Enterprise MDM solution is that enterprises usually require the ability to provide multi-form MDM support for the management of master data. Depending on the methods of use, different MDM implementation styles can be followed. ERP systems and associated scenarios are just one stakeholder of a global Enterprise MDM approach, and typically the integration of master data between operational enterprise systems is the most challenging one. There is no one size fits all approach, a multi-form MDM solution typically involves different technical concepts and associated software components, where some of them may run a central MDM hub and others may run as part of the applications. As mentioned earlier, many MDM approaches simplified and focused on consolidation or registry approaches for specific business scenarios in the past and by doing so delivered major value for collaborative and analytical scenarios. However, when realizing operational MDM scenarios one is specifically faced with the challenge of making the decision regarding centralizing vs. decentralization of MDM functionality. Furthermore, the requirements for integration are much higher. As described by (Dreibelbis et al., 2008), one typical key decision to be made is if either the ERP UI or the MDM UI is used for creation and maintenance of master data. As mentioned above, not much work has been done in the context of identifying best practices for implementing operational MDM solutions in the context of complex ERP landscapes.

MDM Architecture Patterns

According to (Dreibelbis et al., 2008), an MDM architecture captures the best practices for implementing an MDM solution. The MDM architecture is comprised of a detailed set of architectural views that describe MDM solutions that have been repeatedly built and deployed in a consistent, high-quality, supportable fashion with a cross-industry and industry-specific application. MDM architectures should also consider the various Enterprise Master Data business and technical strategies, master data implementation approaches, and MDM methods of use. Furthermore, it should be described using an MDM reference architecture, technical architecture, MDM architecture patterns, and design templates that, when tailored, solve a class of customer problems. In general, patterns are artifacts that have been used, tested, and successfully proven in the majority of recurring situations. (Dreibelbis et al., 2008) have introduced four different groups of master data management architecture patterns: 1) MDM Hub patterns, 2) MDM information focused application integration patterns, 3) MDM process focused application integration and 4) MDM enterprise system deployment patterns. The MDM Hub pattern is a fundamental pattern and establishes either a transaction, a coexistence or a registry hub. The MDM patterns in the second and third group both extend the baseline MDM Hub pattern, where in the first category the focus is set on information integration and in the second category is set on process integration. Finally, the MDM patterns in the fourth group are related to the various deployment scenarios, where the focus is on the integration of other enterprise systems, such as analytical systems, data warehouses, data marts, enterprise resource planning (ERP), customer relationship management (CRM), product lifecycle management (PLM), and other systems.

Solution Blueprints

Our identified business use case “collaboration in corporate groups” follows implicitly already the transaction hub pattern and implements the hub via a dedicated ERP headquarter system. Central master data is created in the headquarter system and distributed from there to subsidiary systems. The users maintain and process either a subset or all attributes of the master data records through the UI of the existing ERP systems as part of the operational business processes. The key challenge in this business use case is to ensure consistent propagation and replication of the globally defined master data objects and prohibit local changes of globally defined objects in the subsidiary systems. The identified business use case of “heterogeneous system landscapes” may involve different patterns depending on the actual scenario. In our case study, we have seen process-focused and integration focused application integration scenarios. Both business use cases do have enterprise system deployment patterns involved.

Especially in complex environments typically more than one pattern is required when building an MDM solution. (Dreibelbis et al., 2008) have recognized this fact and introduced so-called MDM solution blueprints which are a combination of architecture (pattern) blueprints and business (pattern) blueprints. An architecture blueprint is based on a composition and customization of relevant architectural patterns. Business blueprints are the counterparts of architecture blueprints and represent a composition of business patterns. (Dreibelbis et al., 2008) describe various solution blueprints in the context of product information management (PIM) and customer data integration (CDI) from a industry-specific perspective.

(Dreibelbis et al., 2008) have also recognized that MDM initiatives typically do not follow a green field approach. They introduced MDM integration blueprints that help to address some integration problems and understand key architectural decisions that have to be made. For example the SAP application integration blueprint focuses on the question where the master data is authored with specific focus on user interface integration issues.

Requirements for an ERP-centric MDM approach

The results of our case study have shown that in 50% of the cases ERP systems are used as master data management systems, either in an implicit form or in an explicit form by establishing a dedicated ERP-based MDM hub. This indicates the importance of a smooth integration and transition from ERP to MDM and vice versa. We have identified various requirements for an ERP-centric MDM infrastructure based on our case study insights.

The requirements we have identified can be clustered in the following areas: 1) Data Modeling, 2) Data Management, 3) Lifecycle Management, 4) Application Integration and 5) Non-Functional Requirements. In the context of data modeling it is essential that the MDM implementation provides all necessary modeling primitives that typical fully fledged ERP systems contain (e.g. time dependency of business objects). Furthermore, MDM implementations should be able to rely on existing data object definitions contained in existing repositories of ERP systems. Regarding data management, one key element is the ability to lock all changes to master data objects and instances (a standard feature of ERP systems). This is also an important capability to serve compliance standards (e.g. SOX compliance). Delta handling is another feature that is typically required when large volumes of master data are processed. In the area of lifecycle management, the MDM implementation needs to be able to synchronize deployments with ERP systems. This is especially in complex ERP landscapes where based on one or more developments systems multiple productive systems are served. Application integration has various perspectives, UI and workflow integration aspects are of specific importance. Finally, non-functional requirements include for example aspects like availability, administration, etc. It is important to understand that when introducing a new MDM implementation into an existing ERP landscape (which currently manages master data pragmatically as done in 50% of the cases in our study), people will always compare the non-functional capabilities of the ERP platform with the MDM implementation.

Conceptual Architecture of an ERP-centric MDM infrastructure

From the requirements mentioned above we have derived a conceptual architecture, it is visualized below in Figure 3. The principle idea is that MDM capabilities can be found in three interconnected areas: First, there is a dedicated MDM hub implementation which provides consolidation and harmonization. Second, there is an integration middleware which includes an embedded MDM logic for transforming and distributing master data objects. Third, ERP systems should be enriched with an embedded MDM component providing the required core functionality for managing master data locally. Key of this approach is that the three elements share MDM capabilities, e.g. a common and rich model being able to represent and model all necessary characteristics and attributes of master data objects and the lifecycle is managed between the three components (e.g. in the case of changes to an object a consistent propagation is carried out).

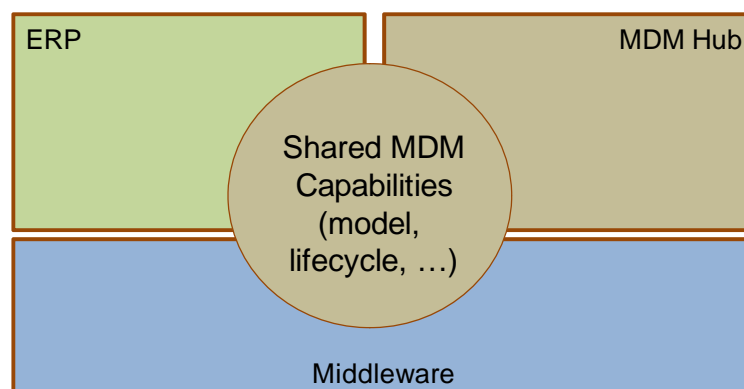


Figure 3. Conceptual Architecture

In an ideal world, both systems, the ERP and the MDM Hub would rely on the same infrastructure including an application server, integration capabilities and business process management. This would simplify the sharing of infrastructure capabilities such as workflow management or user interface components and allow for reuse of corresponding artifacts when switching from the ERP to MDM Hub for master data management.

By following the described conceptual architecture, it would be possible to provide "MDM aware" ERP systems which would allow for deep semantic business object integration with MDM Hubs, which was one key argument of our interviewees for following an ERP-based MDM approach. By providing embedded MDM capabilities already in the design of these systems, it would be possible to expand from an ERP-embedded MDM approach to an enterprise-wide MDM approach relatively easy.

However, the challenge in practice is that ERP systems have evolved over many years and changes in key functions such as master data capabilities are difficult and costly. The current trend of developing and providing more fine-granular Software-as-a-Service (SaaS)-based ERP capabilities may help here. Software vendors should use this opportunity to provide MDM awareness out-of-the-box at least for the fine-granular SaaS ERP capabilities.

CONCLUSION

We have described in this paper an ERP-centric master data management approach. ERP and MDM both target integration, ERP following a business process centric approach and MDM following a broader approach of providing an enterprise-wide single version of the truth of selected master data objects and distribution of them to different target IT systems. Usually every ERP project has been faced with a master data management challenge. Therefore, it is not surprising that ERP and MDM include overlapping master data management functionality. Large enterprises typically do have complex enterprise systems landscapes which typically result in a broader master data management challenge.

We have carried out a multiple-case study analysis at 8 large enterprises to understand the kind of challenges that appear and to identify patterns to approach these challenges within the context of complex ERP landscapes. We have identified and described two major business use cases of MDM in ERP landscapes: 1) collaboration in corporate groups and 2) heterogeneous system landscapes. Based on the identified patterns we have described a conceptual architecture and requirements for an ERP-centric MDM approach. We have put the identified business use cases on the context of the existing work of Dreibelbis et al. (2008) on MDM architecture and solution patterns. The solution pattern "hybrid MDM" has been identified as a good fit for ERP environments and associated use cases. The principle concept behind our approach is that we propose to bring MDM and ERP closer to each other by establishing shared MDM capabilities. This is of specific relevance for operational master data management replication and distribution scenarios, however, also collaborative and analytical scenario can profit from such an alignment.

With the rise of cloud computing and services-based offerings, the enterprise software landscape within enterprises is expected to become even more complex and fragmented. Individual departments or even employees will be able to select more fine-granular pieces of enterprise system functionality in the form of services. This kind of consumption experience will be driven by private life experience, where people can easily combine the different relevant personal productivity services on the Web. We see this today already with the fast adoption of cloud-based offerings in the CRM (e.g. salesforce.com) or HCM (e.g. Successfactors) space. In our future work, we plan to research the question how master data management can be made more light-weight following approaches as described by (Halevy, 2009). Ideally we would be able to empower end-users on the one side, and in parallel, ensure harmonization and governance of master data objects on the other side.

REFERENCES

1. Loshin, D. (2009) Master Data Management, Morgan Kaufmann, Burlington.
2. Knolmayer, G. F. and Roethlin M. (2006) Quality of Material Master Data and Its Effect on the Usefulness of Distributed ERP Systems, ER Workshops 2006, Springer LNCS 4231, pp. 362 – 371.
3. Welch, J and Kordys, D. (2007) Seven Keys to ERP Success, *Strategic Finance*, 89, 3, 40.
4. Berson, A. and Dubov, L. (2007) Master Data Management and Customer Data Integration for a Global Enterprise, McGraw Hill, New York.
5. Dreibelbis, A., Hechler, E., Milman I., Oberhofer, M., van Run, P. and Wolfson, D. (2008) Enterprise Master Data Management: An SOA Approach to Managing Core Information, Prentice Hall.
6. Dayton, M. (2007) Strategic MDM: The Foundation of Enterprise Performance Management, *Cutter IT Journal*, 20, 9, 13-15.

7. Porter, M. E. (1985). *Competitive advantage: creating and sustaining superior performance*. Free Pr.
8. Monk, E. F., & Wagner, B. J. (2006). *Concepts in enterprise resource planning*. Course Technology Ptr.
9. Yin, K (2009). *Case Study Research: Design and Methods*. Fourth Edition. SAGE Publications. California.
10. Vosburg, J. and Kumar, A. (2001). Managing dirty data in organizations using ERP: lessons from a case study, *Industrial Management & Data Systems*, 101, 1, 21-3.
11. Britton, C., Bye, P. (2004). *T architectures and middleware: strategies for building large, integrated systems*. Pearson Addison Wesley.
12. Hohpe, G., Woolf, B. (2003) *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.
13. Bernstein, P.A., Giunchiglia, F., Kementsietsidis, A., Mylopoulos, J., Serafini, L., Zaihrayeu, I. (2002). *Data management for peer-to-peer computing: A vision*. Proceedings of the Fifth International Workshop on the Web and Databases.
14. Halevy, A., Rajaraman, A., Ordille, J. (2006). *Data integration: The teenage years*. Proceedings of the 32nd international conference on Very large data bases. p. 16.
15. Loser, C., Legner, C., & Gizanis, D. (2004). *Master Data Management for Collaborative Service Processes*. International Conference on Service Systems and Service Management, Beijing.
16. Halevy, A.H. (2009). *User-focused Database Management*. Proceedings of the 13th International Conference on Intelligent User Interfaces. Sanibel Island, Florida, USA.
17. Hasselbring, W. (2000). *Information System Integration*. Communications of the ACM, Vol. 43, No. 6, p. 33-38.