TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

ANTTON IKOLA

DEVELOPING MASTER DATA MANAGEMENT IN A MULTI-BUSINESS CASE ORGANIZATION

Master of Science Thesis

Examiner: professor Samuli Pekkola
Examiner and topic approved by the Faculty Council of the Faculty of Business and Built environment on 23.2.2018

ABSTRACT

ANTTON IKOLA: Developing master data management in a multi-business case organization
Tampere University of Technology
Master of Science Thesis, 87 pages, 2 Appendix pages
February 2018
Master's Degree Program in Information and Knowledge Management
Major: Information and Knowledge management, Information analytics
Examiner: Professor Samuli Pekkola

Keywords: master data, master data management, data architecture, data governance, entity resolution

Master data management (MDM) aims at creating and sustaining a single organization-wide unified data reference. In an organization which has multiple business units, this aim brings about many challenges. The challenges can be elaborated through the concepts of master data management, data governance and data architectures. Data governance addresses the roles and responsibilities, as well as common policies and procedures related to the creation, utilization, updating and archiving of master data objects. MDM architecture addresses how the data architecture is organized and in how centralized the actual technical solutions can be.

The goal of this thesis was to find out how to develop master data management in a multi-business case organization. The first phase concentrated on finding out how master data can be management in an organization through selected review of current academic literature. The empirical phase consisted of interviewing the organizations' different stakeholders at enterprise and subsidiary levels. This phase explored the question how master data is currently managed across the organization. The third phase was to identify needs, barriers and possibilities to develop different parts, as well as take into consideration the contingencies that enable effective master data management in a multi-business environment. Case study methods were used in order to have the breadth and depth that is required in providing answers to such complex and organization-specific research area.

The main findings of these research are concluded in three themes: different approaches to developing master data management, data governance and master data management architecture. Alignment of different levels of organization, their needs and different strategies, as well as harmonizing the business processes are important complements to the technical architecture. Therefore, MDM should not be treated as a IT problem, and data governance should not be seen as one size fits all solution. One of the most notable suggested actions of this research is, that the case organization should move towards a common enterprise architecture rather than pursuing separate subsidiary architectures and middle ground solution for MDM.

TIIVISTELMÄ

ANTTON IKOLA: Ydindatan hallinnan kehittäminen monen yrityksen organisaatiossa
Tampereen teknillinen yliopisto
Diplomityö, 87 sivua, 2 liitesivua
Helmikuu 2018
Tietojohtamisen diplomi-insinöörin tutkinto-ohjelma
Pääaine: Tiedon ja osaamisen hallinta, Informaationanalytiikka
Tarkastaja: professori Samuli Pekkola

Avainsanat: ydindata, ydindatan hallinta, data-arkkitehtuuri, datan hallinta

Ydindatan hallinnan tavoitteena on luoda ja ylläpitää yhtenäistä dataa, jota voidaan hyödyntää koko organisaatiossa. Tämä on haasteellista organisaatiossa, jossa on useita liiketoimintayksiköitä. Näitä haasteita eritellään tässä tutkimuksessa käsitteiden master data management (ydindatan hallinta), data governance (datan hallinnan prosessit) ja data architecture (data-arkkitehtuurit) kautta. Data governanceen kuuluvat ydindatan hallintaan liittyvät roolit ja vastuut, yhteiset käytännöt ja prosessit, kuten datan luominen, käyttö, päivittäminen ja arkistointi. Ydindatan arkkitehtuurimallit vastaavat kysymykseen siitä, miten data-arkkitehtuuri järjestetään ja miten keskitettyjä tekniset ratkaisut voivat olla.

Tutkimuksen tavoitteena oli selvittää, miten ydindatan hallintaa voidaan kehittää monen yrityksen case-organisaatiossa. Ensimmäisessä vaiheessa selvitettiin, miten ydindataa voidaan hallita tekemällä valikoiva kirjallisuuskatsaus. Empiirisessä vaiheessa haastateltiin organisaation asianosaisia emoyhtiössä ja tytäryhtiöissä. Tässä vaiheessa selvitettiin, miten ydindataa hallitaan case-organisaatiossa. Kolmannessa vaiheessa tunnistettiin aineistosta tarpeita, esteitä ja mahdollisuuksia kehittää eri osa-alueita, ottaen huomioon erityisiä tekijöitä, jotka mahdollistavat tehokkaan ydindatan hallinnan monen yrityksen ympäristössä. Tutkimus toteutettiin case-menetelmillä, jotta saatiin riittävästi laajuutta ja syvyyttä monimutkaiseen. Tämä menetelmä valittiin, koska tutkimusalue nähtiin monimutkaisena ja organisaatiokohtaisena.

Tutkimuksen päälöydöt tiivistettiin kolmeen teemaan: eri lähestymistavat ydindatan hallinnan kehittämisessä, data governance ja ydindatan arkkitehtuuri. Yhtenäiset linjaukset organisaation eri tasoilla, yhteensopivat liiketoiminta- ja IT-strategiat, yhteiset tarpeet ja liiketoimintaprosessien harmonisointi löydettiin tärkeiksi seikoiksi täydentämään teknisiä ratkaisuja. Ydindatan hallintaa ei tulisikaan kohdella pelkästään IT-haasteena, eikä data governancea tulisi nähdä ratkaisuna joka toimii samanlaisena joka organisaatiossa. Yksi huomionarvoisimmista toimintasuosituksista tässä tutkimuksissa on, että case-organisaation tulisi kehittää organisaation yhteistä arkkitehtuuria sen sijaan, että kehitettäisiin tytäryhtiöiden omia arkkitehtuureja ja luotaisiin välimallin ratkaisuja ydindatan hallitsemiseksi.

PREFACE

This thesis was a big project that seemed first like climbing on top of a mountain. In that perspective, the research journey was adventurous, and I did not know what I was going to find. This thesis represents the last assignment of my studies at Tampere University of Technology. The journey that led up to this leaves me with great memories from the world of academia. As a process, the thesis has been a learning experience, that I am unlikely to experience any time soon.

I want to thank the instructor in the case organization for providing me with insights that helped me with the research. I am grateful also for the examiner of this thesis for providing useful feedback during the writing process. Also, I want to thank the case organization's all stakeholders at the enterprise and subsidiary levels for enabling this thesis, taking part in interviews and for letting me take a look inside their professional worlds for a brief moment.

Finally, I want to thank my colleagues at Columbia Road who encouraged to take some time off work and dedicate my time to complete this thesis. I will not also forget the kind staff at Tampere University of Tampere, who helped me greatly in the last critical days before graduation deadlines.

Tampere, 23.2.2018

Antton Ikola

TABLE OF CONTENTS

TERMS AND ABBREVIATIONS

CRUD	Set of actions that are performed to data. Creating, Reading, Updating and Deleting.
ERP	Enterprise Resource Planning software.
MD	Master Data. The critical data objects and their related metadata, attributes, definitions, roles, connections and taxonomies, which are shared across business areas in the organization (Loshin 2010, p. 6).
MDM	Master data management. A collection of best management practices to organize key stakeholders to incorporate different business applications, data management methods and tools in order to implement policies, procedures, services and infrastructure to support the "capture, integration, and subsequent use of accurate, timely, consistent and complete master data" (Loshin 2010, p. 8).
DG	Data Governance. "A system of decision rights and accountabilities for information-related processes, executed according to agreed-upon models which describe who can take what actions with what information, and when, under what circumstances, using what methods." (Mosley 2010)
Registry style MDM	An architectural solution to manage master data. A reference table or an index of data which links different source data with a global master data key (Allen & Cervo 2015)
Hybrid hub	An architectural solution between the 'thin' registry model and the 'heavier' transaction hub which provides a shared model to manage the identifying master attributes of the data (Loshin 2010).
Transaction hub	An architectural solution which is a single centralized repository that is used to manage all aspects of master data (Loshin 2010, p. 168)
ER	Entity resolution. The process of record linking, data matching or deduplication by sorting out if data objects from multiple source systems refer to the same real-world entity. (Talburt & Zhou 2015)
Enterprise	Used in this research to refer to the case organization which consists of multiple companies
Subsidiary	Used in this research to refer to the sub-company of the case organization.

1 INTRODUCTION

1.1 Research motivation

Data is an important asset to an organization, as it forms the information and knowledge that is needed in order for organizations to compete and succeed (Allen & Cervo 2015). The critical business data in the organization is called master data (Loshin 2010, p. 9). Common examples of master data are customers, suppliers, products and parts (Loshin 2010, p. 6). Managing the key company information has always been important, because it is essential for any company to know, what products they offer and who their customers are, for example. (Loshin 2010, p. 10–11) However, in order to utilize the organizational data, organizations need to clearly define the way data represents the business concepts, integrate the data into a consistent view and make it available across the organization (Loshin 2010, p. 2)

Master data represents a huge challenge for organizations who have developed their data architecture over many years. Organizations might have grown organically or through acquisitions, addressing different line-of-business needs with separate applications, which has led to information siloes (Dreibelbis 2008), and consequently to substantial data management issues (Allen & Cervo 2015). Inconsistent information due to different conceptions and policies might have led to "islands of information" across the organization. This kind of siloed data architecture, lack of common policies and procedures, redundant data and quality issues cause many problems and inefficiencies in utilizing fully the organizations data and information to support business objectives. (Loshin 2010, pp. 1–2; pp. 71)

To overcome inefficiencies due to disparate information structure and eventually create organization-wide business value, organizations need to identify and manage the master data which is used across business areas (Loshin 2010, p. 10). Master data management (MDM) is about creating a consolidated view of the data, "a single version of the truth", which is distributed across the organization (Loshin 2010, p.10). Master data is used across various applications and utilized in different functions of the organization such as procurement, manufacturing and sales (Loshin 2010, p.10). According to Otto & Ofner (2011, p. 1), many software vendors offer MDM application systems, but the user community feels a gap between their own strategic requirements and functionality offered by the software products. Even though MDM solutions present the problem of aligning business with functionality and addressing various stakeholders needs, the usual benefits of master data management include consistent reporting, improved operational efficiency and reduced costs, quicker results, improved business productivity and decision making (Loshin 2010, pp. 11–12).

Master data management might sound like a technical term, implying an IT-driven approach. Nevertheless, the most MDM challenges relate to organizational and governmental issues (Radcliffe 2007, p. 2). This differentiates MDM from IT-driven initiatives such as customer relationship management (CRM) or business intelligence (BI) programs (Loshin 2010, p. 13). Data is residing within technological applications but on the other hand, created and changed with business processes. This calls for attention to both technical and business orientations (Allen & Cervo 2015). In organizations where IT has owned data management, it might be the case that business processes have not taken responsibility of data quality.

IT-driven projects usually imply large budgets, little oversight, long schedules and few early business deliverables (Loshin 2010, p. 13). As such, MDM represents solution independent of specified applications to manage the organizations core data and distribute it across different various IT systems (Maedche 2010, p. 1). Besides managing the data models and quality, MDM is also about data governance, defining policies and procedures, as well as roles and responsibilities considering each data set (Loshin 2010, p. 9). In a distributed organizational environment, designing a master data architecture, assigning roles and responsibilities, designing maintenance and monitoring processes present a complex and multidimensional project to implement. According Loshin (2010) as well as Allen & Cervo (2015) MDM should be started from small set of data which delivers business benefits fast and scaled incrementally across organization.

1.2 Research background

Allen & Cervo (2015) claims that a MDM program calls for dynamic and flexible alignment of business and IT functions and assigning collaborative data management roles and responsibilities to employees in both business and IT functions. An MDM program might also imply establishing a centralized function to manage the master data items, which are used across different companies. Nevertheless, it is imperative for MDM programs to start from small, and from the perspective of addressing actual business needs in the organization (Allen & Cervo 2015).

The case organization is a parenting company, functioning as a management and consulting company for a few subsidiary manufacturing companies. The organization has grown organically through efforts in research and development and exporting, as well as through acquisitions of small and medium sized manufacturing companies. Thus, the organizational structure is fragmented. Some of the acquired companies have implemented enterprise resource planning (ERP) systems at a different time and they have used it with different practices, without common policies, governance or assessing data quality, thus leading to siloed information structure, lacking common data models and procedures, which have ultimately led to poor data quality from organizational perspective.

The organization started to utilize business intelligence (BI) tools in 2013, which made the consolidation of data from disparate information systems easier. It also allowed a consolidated view into the companies' data residing in ERPs. Nevertheless, the implementation of the BI tools revealed inconsistent data models across companies, duplicate data resulting from differences in the way new data objects are created across companies. BI implementation raised a question about the differences in underlying data creation, update and maintenance practices between the companies. It can be concluded that consolidating the data from different companies presents major challenges in both operational and analytical uses of the data for the organization.

The case company has grown through acquisitions and consequently experienced huge data management challenges. Thus, the motivation for this research is to find out how the organization can guarantee the appropriate quality of master data quality in future. Initially, the master data management efforts were seen as a way to make reporting and procurement more efficient, increase communication among subsidiaries' inventories, thus avoiding extra testing for the same components, and possibly transferring components among sub-companies instead of extra orders. Thus, master data management represents also a huge possibility for the organization.

1.3 Research problem

The main research question is "how to develop master data management in a case organization?". This research question can be divided into four sub-questions.

1. How master data can be managed?
2. How is master data currently managed in the organization?
3. What are the barriers to developing master data management?
4. How to develop different parts of master data management in the organization?

The research problem is answered by delivering an overview of the organization's current situation, reflecting the situation to current scientific and management literature, and providing necessary guidelines for further MDM development. Currently, the different parts in which to develop master data management are roughly divided into three sections: master data governance, data architecture and entity resolution.

1.4 Research target and scope

Target for this research is to create guidelines for further development of master data management which are relevant for the case organization. Master data management includes multiple perspectives such as defining the master data elements and common data standards for them, considering different architectures, and clarifying roles and responsibilities (Allen & Cervo 2015). Furthermore, the research target is to create a set of recommendations and guidelines for establishing and maintaining a master data in the organization from the perspective of the organization as a whole but also supporting the needs of the subsidiaries. The guidelines will be achieved using multiple research perspectives presented in master data management literature and research.

The target was limited early on, further governance, architecture, and master data maturity model. The guidelines which are suggested by the literature will be empirically tested (if they exist already in the company). This thesis aims to conclude practices and policies to achieve an efficient level of master data management. Besides some details considering entity resolution, the technical side of master data management is mainly left outside the scope of this thesis. This due to the fact that ETL (extract-transform-load) processes for consolidating data from different sources are already known in the organization, and the main problem at this phase of master data maturity was how to guarantee uniform master data across different business units representing points of master data creation.

The scope of this research is master data management relating especially to components bought by the three subsidiaries. Other data domains such as customer and vendor data, as well as different data types such as transactional data are not included in the scope of this research. These limitations are applied due to the case organization's needs. It is assumed that the organization's future needs call for unified and consistent management and governance practices as well as establishing a MDM architecture. The main problem of inconsistent component data concerning organizations MDM seems to arise from the lack of uniform processes, data governance, clear roles as well as responsibilities in the organization, thus limiting the scope to data governance, data architecture and entity resolution.

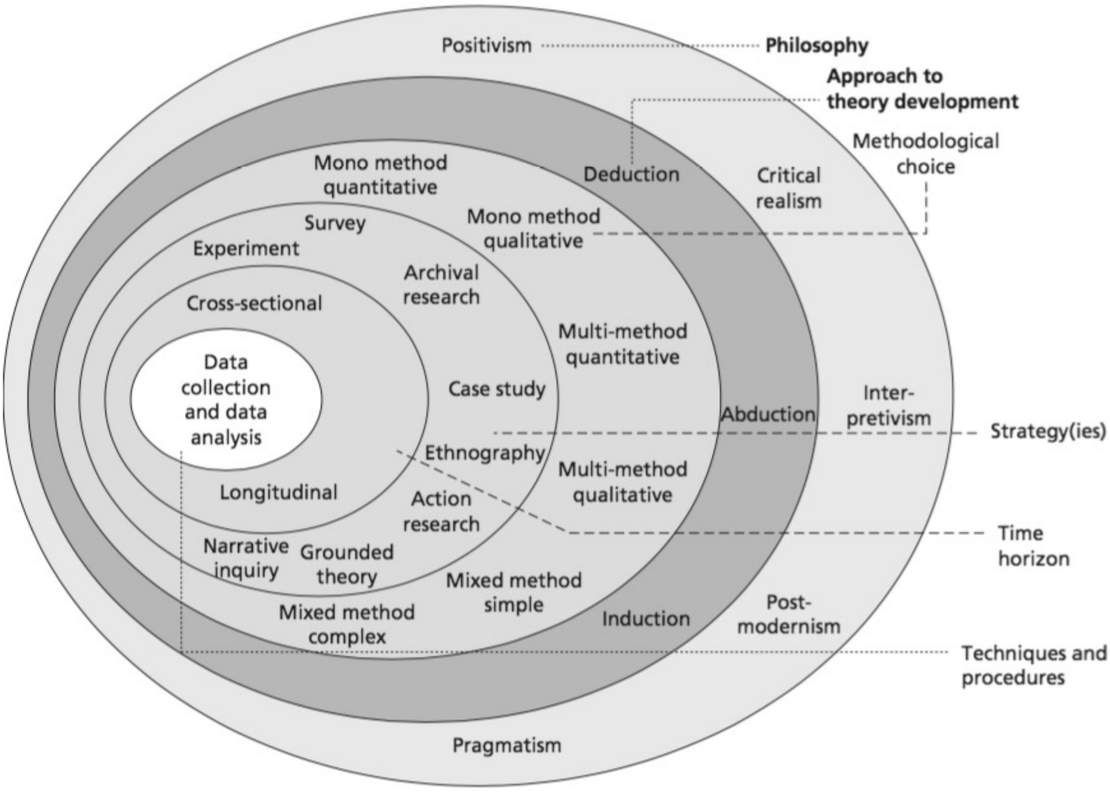
Initially, the operational business needs of the organization hinted that each component data ought to be created in the company ERP in a unified way. The thousands of component instances which reside across separate ERPs need to be consolidated and harmonized eventually. Nevertheless, the total consolidation does not represent the most urgent goal of the MDM initiative. Consequently, the ways to consolidate the already existing product data is not the main focus of this research. Implementing master data management thoroughly requires also planning metrics, training and communication, and forming a road map for development (Pekkola & Vilminko-Heikkinen 2012) which are not included in the scope of this research. In conclusion, the target of this research is to draw guidelines to develop master data management through the processes of defining master data, master data governance, and appropriate architecture, to effectively distribute master data and resolve entities across subsidiaries.

1.5 Research methodologies

It is important to note that every stage of the research includes assumptions which affect the research questions, process, analysis and interpretations. These assumptions are made about what constitutes as human knowledge (epistemological), realities encountered (ontological) and about the extent and ways the researchers own values influence the research process (axiological). (Saunders et al. 2016, p. 124) Saunders et al. (2016, p. 127) note that ontological assumptions (the assumptions about the nature of reality) shape the way we try to solve problems and gives an example about seeing organizational change resistance as phenomena that helping organizations to focus on the most problematic parts of programs, rather than trying to look for ways to completely eliminate resistance. This ontological dimension fundamental to this research also in the case of seeing data as an asset rather than a commodity or result of business operations.

Epistemological concerns of what constitutes as legitimate knowledge range from facts to interpretation and imply a great choice of methods in business and management studies. It is important to note, however, that different epistemological assumptions such as positivism, might imply a specific research method such as quantitative approach. However, if a rich and complex view to the organizational realities is wanted to be achieved, another set of assumptions should be considered instead of positivism. It is likely that this kind of research will not be generalizable. (Saunders et al. 2016, p. 137)

Different research approaches can be represented as a layered onion depicted in picture 1.1., which is “peeled from the outer layer” i.e. approached from the top-level of philosophy to the core and specifics of actual data collection and analysis (Saunders et al 2016, p. 124). The different choices in each level reflect the underlying assumptions of the research.



Picture 1.1. *The research onion (Saunders et al. 2016, p. 124)*

Philosophical choice of the thesis is pragmatism, which is ontologically complex and rich, and takes into consideration processes, experiences and practices. It states that the reality is the practical consequence of ideas and that knowledge has its practical meaning in specific contexts. Theories which enable successful action are considered epistemologically true in pragmatism. Thus, it is a value-driven approach which concentrates on solving problems and developing an informed future practice as contribution. (Saunders et al. 2016 p. 137)

The organizational research paradigm in this research is functionalist, which means developing a set rational explanations and recommendations within the current structures. (Saunders et al. 2016, pp. 130–133). The research problem is seen as one which lacks regulation, and should be solved in somewhat evolutionary style, rather than a radical change. Furthermore, a holistic approach is used. This is due to the reason the aim of the research is to provide guidelines to address organization's needs, and therefore sub-companies are treated as parts of the bigger organization.

The chosen approach to theory development is abductive. This means using both deductive and inductive approaches (Saunders et al. 2016, p. 146). First of all, the deductive approach is used to move from theory to data, to explore the phenomenon through academic literature and to identify themes and patterns. Then inductive approach is used to locate patterns or themes found in the data that correspond to the conceptual framework.

Abduction approach means moving back and forth between theory and data. (Saunders et al. 2016, p. 145) The abductive process of this research is somewhat following: noticing the problem, developing some sense of theory about the causes behind the problem, gathering data about the parts that theory implies, and deducing possible guidelines and recommendations for the organization to test.

The research design consists of a qualitative approach. This is approach is chosen over the quantitative, in order to make sense of the subjective realities of the subsidiaries' different stakeholders. Quantitative approach is not utilized, because it is not seen justified to measure phenomena numerically due to a limited number of stakeholders. Furthermore, the purpose of this research is to find out what is happening and understand the context. Thus, the research purpose can be called exploratory (Saunders et al. 2016, pp. 175–176).

Case study methods are usually used to inquire deeply into a selected phenomenon within real-life setting (Yin 2008). The case study method has been found useful in settings where there are organizational and social issues associated with implementation information systems (Darke et al. 1998). The case study method is seen justified because the research problem of developing master data management implies some change processes in policies, practices, responsibilities and roles. It is important to note that the results of single-case study have imitations with regard to replicability and generalizability (Lee 1989).

The research is a single case study which according to Saunders et al. (2016, p. 186) is usually a sound approach in situations where the problem is especially unique. In this research the single case approach means considering the organizations perspective as an embedded unit. Multiple case approach could be utilized in this kind of research to individually research the individual development of master data management inside the subsidiaries. Yin (2008) also differentiates case studies into holistic or embedded by the unit of analysis. Although the research is a single case study, there are two different units of analysis, and therefore this research represents an embedded case study. The enterprise, or parenting company represents one embedded unit of analysis, and the subsidiaries are the second embedded unit of analysis.

The time horizon of the research is cross-sectional, which implies a snapshot of the current situation across the organization. Nevertheless, the research takes in account the organizations historical aspect of the problem, as well as aims at creating some vision and actions for addressing the problem in future.

Methods for analyzing data in this research are thematic. According to Saunders et al. (2016, p. 579) thematic analysis is a foundational method for qualitative analysis. The purpose thematic analysis is to search for themes or different patterns that occur across data set. The basis for this search is researchers own codifications of qualitative data related to the research question. It can be used to comprehend large amounts of data, identify the key themes, produce thematic description and to draw and verify conclusions. (Saunders et al. 2016, p. 579) Thematic analysis frequently goes further than merely organizing data by interpreting various aspects of the research topic.

2 MASTER DATA MANAGEMENT AND GOVERNANCE

2.1 Data as an asset

Data is the raw substance of information, knowledge, and understanding (Ackoff 1989) and as such, it should be considered a critical strategic asset of a company (Bollinger & Smith 2001). Data should not be considered merely as a commodity or a product residing in enterprise information systems, but as an asset and a resource, which is used in daily operations to perform operations efficiently (Ladley 2012). Data has been traditionally considered through DIKW hierarchy, which states that increasing understanding and connectedness turn individual data points into information, knowledge, understanding and wisdom (Ackoff 1989). Tuomi (1999) contrasts this view by reversing the hierarchy and stating that data emerges only after organizations already have knowledge of the socially shared practices of how to actually utilize the data.

Data has local operational uses in different business areas, but it can also be utilized in analytical purposes to deliver insights and support decision making purposes on global level (Loshin 2010, p. 10). Data management aims to meet the information needs such as availability, security and quality for all stakeholders in the organization (Mosley 2010). Although, it is necessary to emphasize, that different types of data have different purposes and also imply different practices for managing their quality. Data types can be classified roughly into four categories: master data, transactional data, reference data and metadata (McGilvray 2008).

Master data represents the most critical data for organizations (Loshin 2010, p.6). It is data which describes the so-called business objects, such as people, places, and things which are critical to organization's business (McGilvray 2008). The volume and variety of different stakeholders and applications that utilize the information make certain business objects critical. Common business objects are represented in the organization's information systems such as master data. This data is represented by a high degree of reuse and complexity, as it represents the common data objects and related metadata, which are shared across businesses. (Loshin 2010, p. 6) Examples include customer data which can be used in sales and marketing, and product data which can be used in procurement, manufacturing, and reporting functions (Loshin 2010, p. 10).

Transactional data is the data which is associated with or resulting from business transactions, from the concrete internal or external events or transactions that take place when the organization acts in its business. Examples of transactional data include financial data such as order, invoices and bookkeeping entries. Transactional data is usually linked to master data objects. For example, invoices which represent transactional data might refer to the same product objects which master data objects which constitute as master data and can therefore be used in sales reporting and analytical purposes. (McGilvray 2008).

Reference data is sets of values or classification schemas that are used by different systems, applications, processes, reports and by transactional and master records (McGilvray 2008). Reference data can be used to classify or categorize different data, such as master data. Reference data instances can be also seen as master data, when it represents data models shared across different business areas (Allen & Cervo 2015). Reference data management is sometimes used interchangeably with master data management (Allen & Cervo 2015), for example in the DAMA guide (Mosley 2010). The interchangeability is justified "because reference data can be seen as master data when it is shared across organization, and it should meet all the quality standards which are expected from a master data" (Allen & Cervo 2015).

Metadata is "data about data", representing information about the data entities and elements such as labels, usage, changes, type, definition, structure and linkage (Allen & Cervo, 2015). Metadata is utilized in order to make other types of data easier to retrieve, interpret or use. Metadata can be further divided into three categories: technical, business or audit trail uses. (McGilvray 2008) Metadata is an important part of master data as well, as it provides the master data contextual elements by which it can be interpreted (Dahlberg 2015).

Noting the different types of data is important, especially in the context of this research, which focuses on master data. Other types of data related to master data objects are important, as they can be utilized to manage master data more efficiently. For example, reference data can be used to share master data more efficiently, and transactions across different business units are performed efficiently when common master data objects are in place (Allen & Cervo 2015). Loshin (2010) further differentiates master data into three concepts: master data class, master data attribute and master data object. These conceptualizations imply that reference data such as a shared data models can be also seen as a type of master data classes.

To be an asset, any data must be of good quality. What determines good data quality is dependent on the business context and application utilizing the data (Wand & Wang 1996) and usually, data quality is evaluated on basis of "fitness for purpose" (Haug & Arlbjørn 2011). Poor data quality is common, and it is an area which companies have not given adequate attention (Marsh 2005). Business processes, customer expectations, source systems and compliance rules are constantly changing, and these changes should be reflected in data quality management systems and procedures. Data quality is an important topic in itself but will not be covered as a separate chapter in this research, as the focus is in the complexities of developing the MDM function, practices and architectures in MBO context. Even though data quality is a core issue, it cannot be solved merely by defining data quality standards.

2.2 Identifying master data

Master data is usually defined as a set of data which represents critical business objects and entities and it represents the organizations key objects such as: customers, products, parts, suppliers, vendors, locations and accounting items (Loshin 2010, p. 6–7). The concept of each master data object needs to be defined clearly in the organization, so that the responsibilities regarding its quality can also be defined, and the data quality maintained through the business processes that create it, and not left only to IT operations (Brou et al. 2016b). Allen & Cervo (2015) define master data as the most critical data to organizations operations and analytics. Seven main features of master data, summarized by Vilminko-Heikkinen and Pekkola (2012) from literature include: stability, complexity of use cases, reuse across different business areas, high value to organization in general, life cycle of many actions, independence from other data objects, and behavior related to transactions.

Master data can be identified by the multiple business areas, processes and applications that utilize it. For example, product part data can be relevant to research and development, procurement, purchasing and manufacturing. Customer master data is a common starting point to organization's MDM efforts (Silvola et al. 2011, p 148). MDM can also be started from other important data domains, such as product data, but it is important to note, that certain types of data, especially product data can be far more complex than customer data (Silvola et al. 2011, p. 150). Identifying the master data elements can be challenging due to the fact that master data terms such as 'product' or 'customer' have no definitions, or they are ambiguous in the organization (Vilminko-Heikkinen & Pekkola 2012).

An important characteristic for master data is to be referenced in both transactional and analytical system records, such as product management and resource planning systems (Loshin 2010, p. 132). Operational MDM integrates operational applications, such as ERP, CRM and supply chain management in upstream data flow, while analytical MDM reminds data warehousing activities such as customer data integration and financial performance management (Silvola et al. 2011, p. 148). Master data might have specialized application functions for managing the creating, reading, updating and deleting of instances. The master data objects should have a common hierarchical taxonomy, i.e. a reference model, and they are usually managed separately from other types of data. (Loshin 2010, p. 132)

Besides master data possessing certain aforementioned characteristics, identifying master data consists of two main activities: reviewing enterprise data models and evaluating enterprise data assets. Reviewing and redefining enterprise data models is a top-down process, in which critical data objects are identified for business processes. The second part, evaluating the data assets is a bottom-up approach, in which the applications that use the data structures are recognized. In other terms, identifying master data can begin from defining common data structures, and implementing them in a top-down manner, or in a bottom-up manner, starting from existing data structures and resolving them into a common master data environment. (Loshin 2010, p.131)

2.3 Managing master data

"Master data management (MDM) is the application of discipline and control over master data to achieve a consistent, trusted, and shared representation of the master data." (Allen & Cervo 2015). The essential parts of MDM are setting governance policies and responsibilities, defining common data standards and monitoring the resulting data quality (Loshin 2010, p. 9). Master data management comprises of the interactions between data, processes and information systems (Silvola et al. 2011). At the technical level, master data management can be summarized as the processes to consolidate different instances into a unique representation (Loshin 2010, p. 45). On the level of organizational activities, this implies recognizing common data sets and reorganizing them into consistent and current company-wide master data (Loser et al. 2004).

A need for MDM typically arises in a situation where data is duplicated, fragmented, and inconsistent across multiple sources (Allen & Cervo 2015). Consistency and immediacy represent the most general master data challenges (Loser et al. 2004). According to Radcliffe (2007, p. 2) interest in MDM has grown from the same needs that customer and product information management initiatives. MDM addresses the need to create and sustain an organization-wide single version of the truth, a unified data reference, which is utilized in different applications and across business units (Loshin, p. 9-10). This allows for better information quality, integration of different systems, better business productivity, better spend analysis and planning, consistent reporting and improved decision making (Loshin 2010, p. 11–12). MDM has a high impact on business, and thus MDM processes are usually organized at the enterprise level (Reichert et al. 2013).

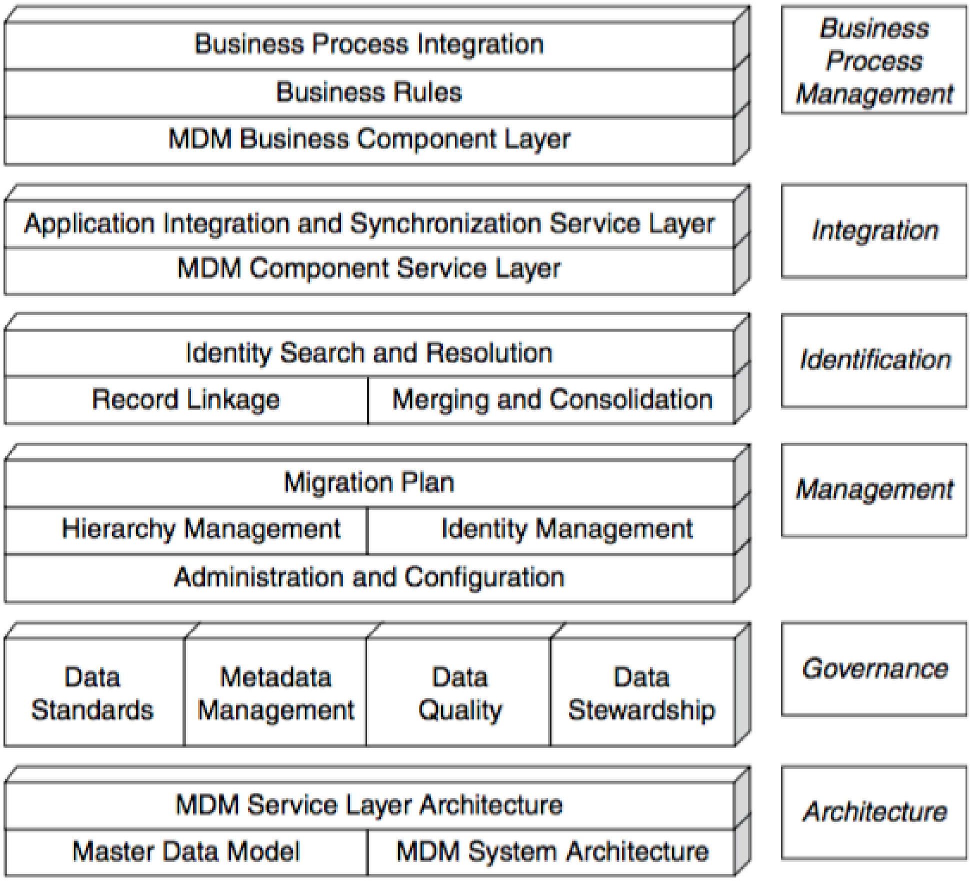
Silvola et al. (2011) note the problems related to MDM on levels of data, processes and information systems. When master data is defined unclearly across the organization, it results in poor data quality (Silvola et al. 2011, p. 157). Suggested response to the problem is to recognize relevant business data, map the current state of data and create a data model which supports company's business objectives (Silvola et al. 2011, p. 157). Problems regarding the data processes are ambiguous ownership of the data, incoherent and non-existent data management practices (Silvola et al. 2011, p. 157). Such problems call for monitoring and continuously improving the data quality as well as modeling the process for data life cycle (Silvola et al. 2011, p. 157).

MDM itself is not the end objective, but rather a mean to achieve business goals. MDM programs need to demonstrate ongoing value creation through a set of metrics (Radcliffe 2006). Business goals that can be leveraged by master data management include for example consistent reporting, improved risk management, decision making, better spend analysis, increased information quality, improved business productivity, and simplified application development. (Loshin 2010, p. 238) Many MDM products lack maturity, fail to resolve problems or attain the business goals. Solving MDM might require combining and integrating multiple applications and products to cover all MDM functions (Allen & Cervo 2015). Such approach has its drawbacks in creating even more inconsistency, and it might necessary to work with multiple vendors to create customized integrations that combine products that were not intended to function together (Allen & Cervo 2015). Multi-domain MDM is thus a difficult task to solve without expert guidance (Allen & Cervo 2015) and it is advisable, that MDM is started from certain business area or an easily manageable set of data (Allen & Cervo 2015; Loshin 2010).

2.4 MDM components

MDM can be seen as a composite of several different areas of components and services, seen in in picture 2.1. The picture describes the various components that are required to align business processes with the actual data architecture and data models. It implies that a functioning MDM system contains all parts in the picture. The organization can examine each components' maturity and begin by adding value by selecting pieces of the model for implementation (Loshin 2010, p. 44). The implementation can be viewed as a bottom up or top down process, beginning from architecture or business process management.

Considering this thesis and the component model, the approach is bottom up, starting the literature review forming the architecture and governance perspectives. Top down approach is utilized in the empirical part which consists of interpreting the implications of current business processes, practical ways to identify entities. These approaches collide, when the implications from literature and empirical part are integrated in the concrete data management practices.



Picture 2.1. MDM component and service model (Loshin 2010, p. 45).

As depicted in picture 2.1. the architecture of MDM can be divided into three distinct components: master data model, MDM system architecture and MDM service layer architecture. Firstly, it is important to choose a domain in which master data is implemented (Allen & Cervo 2015). After choosing a domain, a master data model is created, which implies a unified data model across separate business units and applications. The master data model is created as a centralized effort to create a core resource for any applications utilizing master data. At this phase, the hierarchical model and master data attributes, which are shared globally are defined. All local attributes cannot be included in the master data model.

Vilminko-Heikkinen & Pekkola (2012) state that establishing MDM function is a process consisting of the following stages:

1. Identifying the needs and objectives
2. Identifying the organizations core data and processes that use it
3. Defining the governance
4. Defining the maintenance processes
5. Defining data standards
6. Metrics for MDM
7. Planning a MDM architecture
8. Planning the training and communication

9. Forming a road-map for MDM development
10. Defining MDM applications' functional and operational characteristics

These stages touch upon the Loshin's (2010) components (picture 2.1), and provide guidance to the practical way of approaching MDM simultaneously from the top, the specific business needs and from the bottom, the actual data. Cleven & Wortmann (2010) further elaborate four strategies to approach master data management that are combination of process or data-driven and problem or solution oriented. Problem-oriented strategy represents low effort, but it might lack a systematic approach (Cleven & Wortmann 2010).

MDM system architecture consists of technical components by which the master data is managed throughout its life cycle. The CRUD functions operated on the master data are the basis of this life cycle. MDM architectures and their implications are further elaborated in chapter 3. The second level, data governance means briefly the assigning of roles and responsibilities related to creation and maintenance of master data. This level is elaborated in the chapter 2 concentrating mostly on the organizational roles and responsibilities such as stewardship.

2.5 Master data management maturity

The transitioning to MDM should be considered as an evolution rather than a sudden revolution (Loshin 2010, p. 65). Master data maturity describes the level of an organization's capabilities in terms of master data architecture, governance, management, identification, data integration and business process management. Maturity is also about overcoming the common barriers and misconceptions about master data management. Many barriers deal with the fact that it is not recognized that poor master data quality brings negative effects and thus, roles and responsibilities are not clearly assigned to MDM (Haug & Arlbjørn 2011). The maturity model can be used as a yardstick against which to reflect the current state, as well as to project a desired end state (Loshin 2010, p. 65).

Loshin (2010) classifies master data maturity into 5 distinct categories: initial, reactive, managed, proactive and strategic performance. The maturity model can be used to evaluate the organizations current capabilities and possibilities to further advance master data management. (Loshin 2010, p. 55)

At the initial level of maturity, most of the capabilities to exploit master data are limited or non-existent. In practical terms, there might exist duplicated data sets which are relevant to more than just one application. At this level, business and technical managers look for ways to consolidate sets of data for analytical purposes. (Loshin 2010, p. 55) Ofner et al. (2013) describe the level one in enterprise data quality management as "establishing awareness".

The following stage of maturity is called reactive. It adds capabilities to exploit master data by recognizing replicated copies of the same data and their consequent business issues which are attempted to resolve. Nevertheless, the data problems are still seen mostly as IT problems, and not business problems. The actual business needs for master data have not been analyzed, but IT team might have acquired tools to satisfy some line-of-business' needs. These reactions to bad data in business areas lead to individual solutions and excess duplication of data if the lessons are not shared and if more profound solutions such as consolidating metadata are not used. (Loshin 2010, pp. 55–56) In enterprise data quality management, this level corresponds with "creating structures" (Ofner et al. 2013).

From reactionary it is essential to try to organize the siloed structure and move into managed stage. Master data is now used heavily by analytical applications which rely on further level of consolidation. This allow making value propositions and plans for further use and growth of master data repositories. At this phase, lessons learned from single business area solutions are shared. The ability to use master data evolves into a repeatable process, which can be expanded into new and already existing applications. (Loshin 2010, pp. 56–57)

Proactive stage relies on establishing core data models and service architectures and therefore reduces the dependence on keeping duplicates of data. Additional service layer enables easier integration of applications. Component service layer might include synchronization for application data, identity resolution, as well as hierarchy and identity management. It might also include additional capabilities to establish data integration relationships with customers, suppliers and vendors. In addition to broad consolidation, the service layer might provide also aggregated data as a core enterprise resource. Enabling this stage generally requires that data governance is in effect across the whole organization. (Loshin 2010, p. 60)

The final stage of maturity according to Loshin (2010, p. 62) is the one aiming for strategic performance through rapid capabilities to develop high quality applications to support both operational and analytic requirement of enterprise applications. In terms of enterprise data quality management, this level represents "becoming effective" (Ofner et al. 2013).

2.6 Data governance as an enabler of MDM

Definition of data Governance is "decision rights and accountabilities for information-related processes, executed according to agreed-upon models which describe who can take what actions with what information, when, under what circumstances and using what methods" (Ladley 2012). Data governance and MDM are important parts of larger enterprise information management activities (Ladley 2012) and as such there must be assurance that different lines of business comply with the rules that govern participation (Loshin 2010, p. 68). Lack of delegation of responsibilities is recognized as one of the five main barriers for master data quality (Haug & Arlbjørn 2011). Data governance efforts allow the organization to also gain competitive advantage by enabling effective handling of the data assets (Rindfleisch & Moorman 2001). In essence, data governance is a disciple to manage data for better results by taking various perspectives to the interaction between the people, technologies and data of the organization (Loshin 2010, p. 67). There is no one universal solution for data governance. An organization defines its unique configuration by defining roles, domains and responsibilities, and decides if specialized people need to be hired, trained and integrated into the organization (Brou et al. 2016a).

For MDM, the three most important aspects of data governance are defining practices for managing critical data elements, ensuring the monitoring of information policies, and documenting as well as safeguarding accountability for high quality master data (Loshin 2010, p. 86). The monitoring and oversight provided by a proper data governance framework enables the successful implementation of MDM initiatives, but it should not be initiated in the organization without clear perception of the business value that MDM represents (Loshin 2010, p. 68). The need for organization-wide vision of master data is usually self-evident, but a major risk remains, that the individual views remain as a practice. This hindrance needs to be overcome by demonstrating the effects of common and good enough quality master data, as well as the importance of right practices. (Vilminko-Heikkinen et al. 2016).

First of all, the aim of data governance is to prevent faulty data in the first place (Allen & Cervo 2015). Furthermore, it aims to assess and manage the risks related to enterprise information and to reduce the impacts which are caused by lack of monitoring. These monitoring policies and procedures need to be defined and distributed across the organizations stakeholders. Organization can be prepared for the transition to data governance by asking proactive comments from different application teams, building consensus and defining a stewardship framework to manage the data. (Loshin 2010, p. 68) Data governance can be seen as the most effective way to ensure data quality because it aims to prevent the faulty data in the first place (Allen & Cervo 2015). Data governance represents preventive measures, such as reviewing and specifying data policies and aligning them to reflect business needs and expectations (Loshin 2010, p. 68–69). Although there might be a recognized need for data governance, the tasks and responsibilities are often avoided, especially those which concern organization-wide development. Thus, the ownership of the data can be partially divided, and management teams can be used to share responsibility. (Vilminko-Heikkinen et al. 2016, p. 12)

Ultimately, data governance aims at aligning the master data management efforts with the organizational business management objectives. Thus, the guiding business strategy and its implications for data policies need to be effectively communicated. In conclusion, it must be clearly communicated how data assets are used in the organization and how they are supposed to be managed over time. (Loshin 2010, p. 69) A common underlying data governance framework can benefit separate business units who have a sufficient degree of freedom to make their own resource allocation and data governance decisions while at the same time working in cooperation with affiliated business units. On the other hand, the other end of spectrum, the centralizing master data management might bring forth political issues: reassigning the roles and responsibilities and transitioning into a new process might be experienced as a threat (Loshin 2010, p. 75).

Data governance program can be seen ultimately as a building consensus for commonly defined data, coordination and collaboration in the organization (Loshin 2010, p. 74–75). Ladley (2012) claims that data governance should be initiated through a program, but eventually it should disappear as a stand-alone program, when it has gradually become a part of the organizations daily actions. After all, when data is seen as an asset to increase competitive advantage and not merely an operational commodity, it is evident that governance and maintenance are accepted as normal activities (Ladley 2012). Vilminko-Heikkinen et al. (2016) add that instead of speaking of a MDM project, the project should consist of business cases that point to clear areas of expertise that are used to engage people. This implies taking in account different business units interests and building consensus for data governance practices.

2.7 Data governance framework

Master data governance requires clear roles, stewardships, responsibilities, decision areas and activities (Vilminko-Heikkinen & Pekkola 2012). The data owner's role includes authorizing the creation and maintenance of master data and taking absolute responsibility of the quality and accuracy of local master data. Employees in this role are likely to be approvers of data and may delegate the data further to a provider or to the actual master data repository. The owners do not necessarily maintain their data, but there might exist a different person in charge of creating and maintaining master data in ERP systems. Maintainers work according to data requests, meeting the business expectations for the master data. (Duff 2005) The locus of control in data governance can be positioned functionally in business departments, or in IT department, but shared responsibility between these two is usually recommended (Otto 2011). There is often a clear need to address the ownership of data, but paradoxically the data owners stay committed to group specific functions instead of organization-wide development (Vilminko-Heikkinen et al. 2016).

According to Loshin (2010, p. 82) there needs to be a management structure in place to oversee the execution of governance in addition to a compensation model in place that rewards execution. Responsibilities concerning master data are defined at organizational, support function and actual data set levels (Vilminko-Heikkinen & Pekkola 2012). Furthermore Duff (2005) recognizes three distinctive roles regarding data: the owners, users and maintainers. These seem to correspond to Loshin's (2010) different levels as well. According to Loshin (2010, p. 82) many organizations have appropriate data governance policies but lack underlying organizational structure which would assign responsibility and monitor accountability. Furthermore, Loshin (2010, p. 82–83) divides the specific roles into four following hierarchical categories:

- Data governance director
- Data governance oversight board
- Data coordination council
- Data stewards

The governance framework should support the needs of the whole organization from "the top down and from the bottom up". In practice, this means that the executive sponsorship is also needed to ensure strategic direction, funding, advocacy and oversight (Weber et al. 2009, p. 11). The governance oversight board ensures that the actual data activities meet the required data policies for quality. The coordination council monitors and manages the governance across different business areas and delineates responsibilities and accountabilities to the data stewards, in the deeper levels in the organization. The data stewards follow the data quality criteria for each application in their business area. (Loshin 2010, pp. 82–83)

Data governance director manages the data governance at the enterprise level and is responsible for providing guidance to all participants in daily activities. The data governance director is responsible at the top level that the information policies are in accordance with business needs. The director plans and operates the governance oversight board and identifies the need for new governance initiatives. He is also responsible for providing executive reports on data governance performance. (Loshin 2010, p.83) According to Vilminko-Heikkinen et al. (2016, p. 12), it is important to keep also the executive branch informed, but keep communications minimal, which call for clarifying both data governance and application development objectives.

Data governance oversight board decides the strategic direction for enterprise data governance. It consists of various employees chosen across the organization. The board is responsible of overseeing the current information policies and procedures as well as transforming the organizations changing business needs into new information policies and specific data rules. New data governance policies and processes are accepted by the board, and the related reward framework for compliance is managed. New proposals for practices and policies are reviewed by the board. Furthermore, the board endorses data certification and audit processes. (Loshin 2010, p.84) In short, the governance oversight board regularly reviews MDM performance against set goals.

Data coordination council directs and manages the actual governance activities. It consists of a group of interested stakeholders from across the organization. The coordination council receives and operates under the strategic directions of the data governance oversight board. Council adjusts and oversees the activities so that the data governance expectations are reflected in the actual data quality. Data coordination council oversees the work of data stewards and tasks of any advisory groups related to data governance and provides them direction and guidance. The council leads, promotes and facilitates the governance practices and processes, and thus advocates for enterprise data governance. It also nominates the data stewards and can by itself name and appoint the representatives to data committees and advisory groups. The governance oversight board and coordination council functions can be assigned to a single work group at the initial stage of governance. (Loshin 2010, p. 84–85)

Data stewards can be focused more on specific business areas requirements for MDM standards and policies, or stewards can be technical and provide standardized data elements, definitions, and explain data flows between different systems. Business data stewards can be assigned to functional departments, while the technical data stewards usually function in the professional IT department. (Weber et al. 2009, p. 11).

Furthermore, the responsibilities can be assigned by two main design parameters, presented by Weber (2009) "organizational structuring", which ranges from centralized to decentralized, and "coordination of decision making", which ranges from hierarchical to cooperative. The main difference in organizational structuring is that in centralized data governance design, the data stewards are responsible, but the accountability is on higher levels. In a decentralized model, stewards hold more accountable role. Therefore, in the decentralized model, the data governance director, oversight board and council are mainly consulted, not accountable. (Weber et al. 2009, pp. 14-15)

In addition to the data governance framework that depicts the organizational structuring of data governance, it is important to note that assigning different decision-making roles is also necessary for effective data governance. A responsibility assignment matrix, such as the most popular RACI (acronym for Responsible, Accountable, Consulted, Informed) chart can be used to identify participants and the degree of interaction with certain activities or style of making decisions (Wende 2007, p. 422). This role-based classification can be of great help when assigning data governance in specific domains of data.

Assigning the structural roles as well as the responsibility level is important for the success of data governance implementations. Furthermore, it is important to identify the individuals and groups that can gain most from the development of a specific MDM domain and demonstrate the effect of their roles and related actions to development (Vilminko-Heikkilä et al. 2016, p. 12). In conclusion, the framework should be created according to the needs of organization. The design should be thought through the important parameters and made sure that all activities are clearly assigned with certain domain, role and responsibility.

2.8 Data stewardship role

Data stewardship is a role in data governance framework who is responsible for supporting the data user community by collecting, collating and evaluating issues and problems with data (Allen & Cervo 2015; Loshin 2010, p.85). Modern view is, that data stewardship is a role in between business and IT (Allen & Cervo 2015) overseeing the accountability for business responsibilities and effective control and use of data assets (Mosley 2010). It is the underlying success factor for MDM (Allen & Cervo 2015). Data steward is the key role that spans all domains of master data (Dreibelbis 2008). The data steward's broad responsibilities include "driving the correction of data issues, improving overall data management process, and focusing on the content, context, quality, and business rules which surround the data" (Allen & Cervo 2015).

Wende (2007) presents that there can be three kinds of roles: chief stewards, business and technical data stewards. Technical data stewards reflect the technical needs, business data stewards the business needs, and the chief steward is supposed to consolidate these views (Wende 2007). Whether this elaboration is practical, is not validated in literature, and Wende (2007) notes the roles vary from company to company. Data stewardship role is still generally supported by both IT and business resources that have the necessary knowledge, skills, and focus needed to support and control master data. (Allen & Cervo 2015).

Stewardship is usually limited to a specific business area or subject responsibilities (Loshin 2010, p.85) and the role can be either business data or technical data oriented (Wende 2007). In MDM the usage of data entities might spread across the whole organization, the stewardship role might also be limited to a certain key data domain or element. Stewards should assign root causes behind the data issues and communicate the top priority issues to all stakeholders who are able to solve the issue, or on the other hand, who might be affected by the issue. (Loshin 2010, p.85). Data stewardship should be rationalized through case demonstrations and trainings by demonstrating, what happens in the downstream if the data is of poor quality (Allen & Cervo 2015).

Data stewards should be positioned in-between creators and users of master data, as well as in accordance with the business model of the organization. Stewards are not merely agents promoting data governance and standards. Data stewards need to be closely aligned with the various applications where the data resides and with the users of the master data. By examining the usage and flow of master data in particular data domain, it is possible to determine critical points where data stewardship can be most effectively applied. Data stewards should be in a position where they can most influence data management, data entry, usage and quality control. (Allen & Cervo 2015) Stewardship is neither a full-time position or a job title, but rather a role that has certain responsibilities and accountability towards a business area and the organization as a whole. (Loshin 2010, p.85)

Without a consistent and well-aligned data governance and stewardship in place, a MDM program is unlikely to succeed. Local and functional data governance and stewardship practices need to be taken into account. It is possible that the practices are too narrow in definition, i.e. focusing on local needs, and need to be aligned with an enterprise data governance strategy and plan. The ultimate goal is to support both the enterprise and local requirements. Local order processing might require customer's name, address, and email, but not telephone number, even though other functions in the organization such as marketing could benefit from this information. Well positioned data stewards can recognize and capture these needs and communicate them efficiently. (Allen & Cervo 2015)

In a centralized data hub architecture data stewardship should focus on managing the quality of data coming entering and leaving the hub. In this architectural choice, the data steward is conceptually residing between the master data hub and the different line of business transactional processes. (Allen & Cervo 2015)

In a decentralized model, where the operational domains are handled on a local, regional basis, which implies that stewards need to engage across source systems and other environments. The data may remain unconsolidated across different systems, and still contain different quality and consistency issues. This calls for more mapping and normalization of data in different points. The data stewards should be positioned across various transactional and analytical processes, system areas with the vendor data entry points. This implies that data stewards are needed across multiple regions, and that data stewards should for their own community to address and manage the data issues in the source systems. (Allen & Cervo 2015)

In a federated data governance model, which is more likely than a completely centralized one, the multi-domain nature of MDM is taken into account. An architecture is designed for each data domain's needs, which might mean, for example, that customer and product domains operate in a centralized manner, but the manufacturing domain is decentralized. (Allen & Cervo 2015) This implies that stewards are placed differently across different data domains.

2.9 Factors affecting data governance style in a multi-business organization

It is important to note, that many of the academic ideas of the factors that influence data governance have been inherited from relatively more studied IT and corporate governance. Aligning IT strategy with business strategies remains the top issue in information systems discipline. In today's world, when the hardware and infrastructure are moving towards cloud-based solutions, the data aspect is in the focus. Alignment creates value through competence, governance and flexibility (Reynolds & Yetton 2015). IT governance has been studied extensively in terms of what kind of styles there are, and what affects the choice of governance style. Governance styles can be seen very similar in IT and data governance frameworks, mainly, ranging from centralized, federated and decentralized styles. Various contingent factors can either reinforce, conflict or dominate in respect to each other (Sambamurthy & Zmud 1999).

Each company requires a set of data governance style, roles and processes that fit in their situation. Aligning business and IT strategies between MBO and multiple strategic business units involves dealing effectively with the trade-off between centralization and decentralization (Reynolds & Yetton 2015). In order to lower organizational coordination costs, organizations seek to mirror their corporate arrangements in the activities of their key subunits, which usually implies that organizations with centralized corporate governance tend to centralize their IT governance as well (Sambamurthy & Zmud 1999). Decentralization and placing the responsibility on the business line managers might be compelling, but in such cases, each business unit should have adequate resources to implement the strategies. (Hamel and Prahalad 1989)

The data governance style can be seen contingent on multiple different factors represented in table 2.1 (adapted from Weber et al. 2009). The table presents a framework to assess and evaluate the different contingencies, and their possible consequences for data governance design.

Contingency factor	**Definition**	**Placement of data quality management activities**		
		Centralized	**Blended**	**Decentralized**
Performance strategy	Organization performance objective that businesses emphasize	Profit	Asset utilization	Growth
Firm size	Number of employees, sales	Small		Large
Diversification breadth	Degree of product/market relatedness of a multi-business firm	Related diversification		Unrelated diversification
Organization structure	Degree of centralization of the corporate governance mode	Centralized	Federated	Decentralized
Competitive strategy	Type of engagement in product/market development and commitment to stability	Defender	Analyzer	Prospector
Degree of process harmonization	Level of harmonization between multi-business' processes	Globally harmonized		Locally harmonized
Degree of market regulation	Scope and level of market regulation between businesses	Globally uniform regulations		Highly local regulations
Decision-making style/culture	Informal rules that determine how things get done and what kind of behavior is acceptable	Hierarchical		Cooperative

Table 2.1. *Contingency factors regarding the Data Governance Design (adapted from Weber et al. 2009)*

The organizational structure can be seen as a spectrum: from centralized to decentralized. For MBO's, a centralized organization structure may imply a corporate strategy, and a separate strategy for the business unit. Corporate strategy answers how the MBO can compete as organization, while the business unit strategy answers how the specific unit will compete in individual markets. Aligning these separate strategies is an important, yet, problematic dilemma.

Having separate business strategies might call for individualized data governance but if combining and aligning them closely, the whole MBO and separate business units can center themselves around an IT platform, thus making the data governance and architectural choices a clear competitive advantage. This kind of alignment is possible in situations of related diversification and can create a functionally aligned organization with related lines of business. In case organizations become more diversified, it might be useful to have a MBO lever IT platform to reap the benefits. (Reynolds & Yetton 2015 p. 111)

3 MDM ARCHITECTURE

3.1 Clarifying the current architecture

Loshin (2010, p. 70) states the current data architecture should be clarified in order to notice the overlapping information and assign data governance across the organization accordingly. Important questions for assessing the current data architecture are: what data assets exist, how they are currently managed and utilized and how they support existing application architecture (Loshin 2010, p. 70). Also, the various different master data artifacts should be inventoried, namely: data models, data dictionary, functional architecture (IT-business process interaction), source to target mapping, data life cycle and CRUD analysis. These domains should be inventoried, gathered and reviewed for each master data domain. (Allen & Cervo 2015) Clarifying the data architecture requires a lot of resources and prudence, so it is advisable to focus on the key data entities that have widespread business relevance (Loshin 2010, p. 70). Developing MDM is an evolutionary process, which is why analyzing the current state of architecture is essential.

After the inventory of identifying data sets and enumerating the data attributes within, it is recommended to evaluate where inefficiencies and redundancies create roadblocks for proper monitoring. The collection of data should be reviewed by team of experts and cast into a new logically consistent new view that will be used enterprise-wide. Every time a data element is created, modified or retired, it must contribute to the organizations business objectives somehow. On the other hand, the success and failure of business objectives is related to information activities that support the activity. In conclusion, starting from the data inventory represents a bottom-up approach, moving from data and information functions towards business objectives. In an ideal situation, the key data elements and related attributes and their information policies should be derived from the business objectives in a top-down manner. (Loshin 2010, p. 70–71)

The data quality expectations need to be communicated and monitored clearly so that related business objectives can be effectively met. Data quality should be reviewed by the essential metrics in either periodically by managers or stewards, or constantly between information process stages. Quality expectations should be deployed as validation rules for inspection along the different phases of the data's life cycle, which allows monitoring accountability as well. In a siloed environment, mangers of different business areas are usually assigned as data owners, but the organization level data quality metrics might be ignored. (Loshin 2010, p. 72) The data elements and attributes quality metrics represent the key metrics and imply practices which drive alignment and consistency in each data domain.

3.2 Design principles for MDM architecture

Systematic design of master data management architecture is prerequisite for efficient master data management (Otto & Scmidt 2010). Data architecture design aims at forming unambiguous differences between local and global data, specifications of business-driven data and consistency across a variety of applications (Ebner et al. 2012). To efficiently exchange master data across the organizations different applications, the information systems need to be integrated (Loser et al. 2004). MDM architectures represent the ways in which the data is managed in logical and physical structure (Loshin 2010, p.159), i.e. the functions and tools by which the entities and hierarchies are created and maintained. The different architectural models aim to deliver transparent and shared access to the unique representation of master data and the actual service layer which matches core identifying data attributes to distinguish unique instance from another (Loshin 2010, p. 165).

Oftentimes, the most critical data domains are integrated first to gain business benefits such as cost savings and productivity increases, and improved reporting (Loshin 2010). The most common data domains to implement MDM have been customer, product, locations, finance and employee. For manufacturing companies, the most important domains to integrate are usually customers, products, suppliers, materials, items and locations. (Allen & Cervo 2015)

MDM is supposed to be customized according to the organizations business needs (Allen & Cervo 2015). This notion has risen from the user community of current MDM systems, who experience major mismatches between their own strategic requirements and the functionality currently offered by the most software vendors (Otto & Ofner 2011). Current IT strategies, which MDM is an essential part of, need to support the current business strategies as well as enable future business strategies, while on the other hand, the business should cultivate the development of future business strategy (Reynolds & Yetton 2015). Thus, it is important to note that there are no existing solutions or universal data models, that would fit every organizations business needs, nor architectural styles that would suit every organization and data domain (Allen & Cervo 2015).

Otto & Ofner (2011) conclude 7 design principles for future MDM systems. Master data should be seen as:

- a raw product to produce information in the organization
- recognizing the internal and external market for master data
- subsidizing the MDM processes to smallest, lowest and least centralized competent authority
- being aware of the master data's business context
- describing "the nucleus" of mandatory properties of a master data class
- ensuring the quality over CRUD processes, and
- eventually integrating seamlessly for sufficient interoperability.

There are several differences between MDM styles, of which the first and foremost is the degree of orientation towards physical instantiation or more virtual metadata representation (Allen & Cervo 2015). According to Loshin's (2010, p. 166) terms, the style can be a fully consolidated master data or a virtual master data index, or some hybrid combination of these styles. The style affects latency of interaction with the source systems, where master data is authored (source systems or centralized hub), suitability for direct access by upstream (operational-transactional) and downstream (analytical) applications. The style also affects, whether the data is created and managed centralized or in a decentralized manner in source systems, and if the data is it copied, merely referenced, migrated or updated (Loser et al. 2004). The cost of the implementation also varies, due to varying degree of harmonization and management efforts (Allen & Cervo 2015).

Loshin (2010, p. 166) presents three distinct architectural styles which are tightly federated transaction hub, a simple reference architecture where isolated systems are merely linked by a simple registry and a combination of these two called hybrid centralized master. Loser et al (2004) in turn classifies architectural styles by the degree of centralization and level of common attribute definitions into four distinct categories: central master data system, leading system, repository and common standards approach. Loser et al. (2004) does not present the tightly consolidated transaction hub, but on the other hand, Loshin (2010) doesn't introduce the common standards approach to consolidation. Considering the common standards part, the data governance has addressed this problem, and will not be elaborated in this chapter. Nevertheless, it is seen important to present here certain predecessors to "real" MDM solutions, mainly the analytical use of MDM.

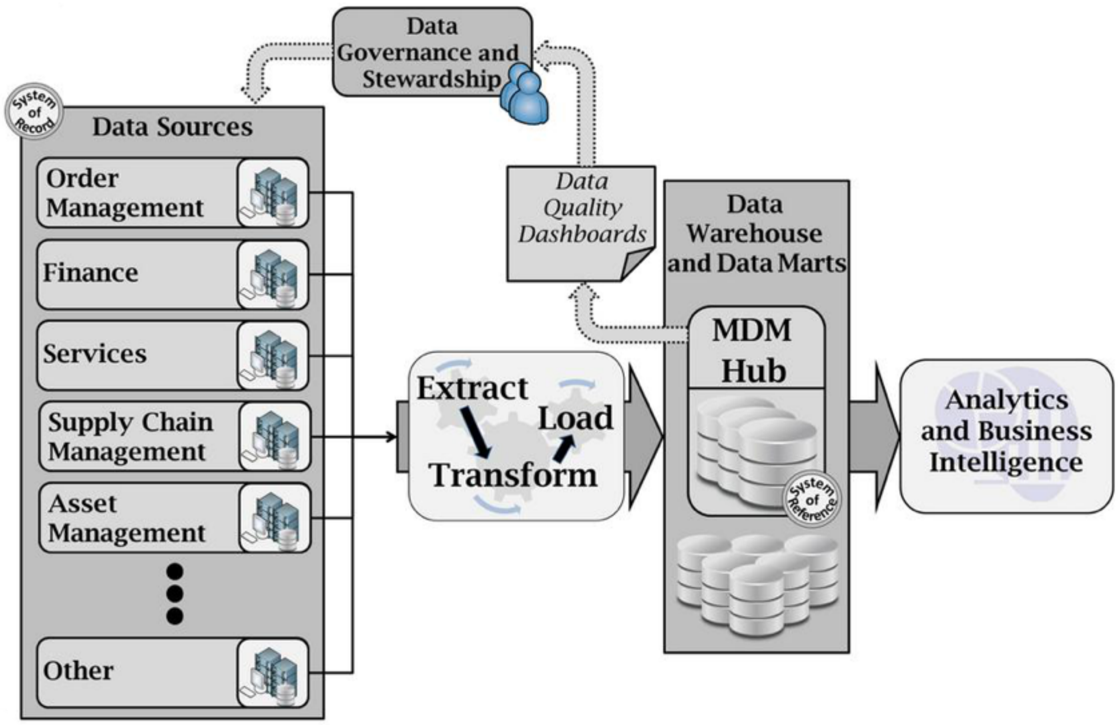
3.3 Analytical MDM as a starting point

MDM implementations often start in analytical usage, which responds to consolidated reporting needs. Analytical MDM is an architecture style which relies on a data warehouse, a business intelligence tool and an operational data store implementation. Analytical MDM adds formal data quality capabilities to cleanse, standardize, consolidate, and augment existing information in the form of extract-transform-load (ETL) scripts. Its purpose is to increase revenue by improving analytics, reports and business intelligence. (Allen & Cervo 2015)

Analytical MDM cannot be considered an enterprise solution because it considers operational systems such as ERP's as read-only environments. Thus, the entity resolution provided by the analytical MDM solution does not affect the operational systems' data quality. Improvements in data quality have to be duplicated in all of the operational systems, which might cause scattered and inconsistent data and several update procedures. (Allen & Cervo 2015)

Analytical MDM can be seen as the first phase of evolvement of aggregating data to support decision-making processes. Nevertheless, it treats the source systems as read-only environments (Allen & Cervo 2015), and therefore relies much on improving the source data quality. This, in turn implies data governance.

The analytical MDM architecture and its information upstream are depicted in picture 3.1. Firstly, the data is loaded from the source systems through elaborate extract, transform and load processes, which can be a complicated process in case of many source systems, and might cause latency issues because the data is processed in batches. The data is classified and further combined in the data warehouse or "MDM hub" in this case. Considering the problem of this thesis, it is important to note that the downstream ETL processes as well as the analytical use of data already enable some level of visibility to the actual data quality. Analytical dashboards, compiled from source system data, might imply data governance and stewardship processes to be instantiated to the data owners by specific data rules.



Picture 3.1. Analytical MDM logical architecture (Allen & Cervo 2015).

In this MDM style, it can be useful to consider if operational environments (depicted data sources in picture 4.1) have possibilities to benefit from master data initiatives. If so, there might be better architectural approaches which can result in win-win for both top management analytics and operational systems. This style represents the bottom-up approach for clarifying data architecture, because it starts with the inventory of existing data to bring the data quality problems visible and applying governance methods to comply to identified business policies where the data rules touch.

3.4 Registry, Hybrid Hub and Transactional MDM

Different architectural styles aim to an appropriate balance between the interoperability and integration in the enterprise (Chen et al. 2008). *Registry style MDM* is in essence a reference table or an index of data which links different source data with a global master data key. The registry style serves to find instances of an entity at their various sources. This style has many benefits: it is relatively low-cost solution and it can be implemented quickly. It has low risk since source data is not edited, and it relies on minimal amount of shared knowledge across systems. Registry style might suit companies that have many data sources and business areas that are sensitive about ownership and changes in data. (Allen & Cervo 2015) Relatively low number of attributes, low need for synchronization and need for high performance are some implications to select registry style architecture (Loshin 2010, p. 173).

A simple registry might provide an entry level consolidation for few attributes or master data objects (Loshin, p. 169), but the downside of registry style is that if source data itself is of poor quality, there are not that many benefits from consolidation. The entity resolution provided in the registry style can be either static, which requires separate federation and updating, or dynamic, which requires automated matching algorithms. Any automated matching algorithm produces many false positives and false negatives. On the other hand, the static list, requires creating a robust metadata model consisting of global master data keys. Only the global keys which link the data are included in the hub, leaving out any duplication of data. Therefore, it is critical that the metadata list is constantly upgraded to reflect changes in the source data. (Allen & Cervo 2015)

The global keys are stored centrally, but the authoring and updating of the actual data object remains distributed in the source systems (Loser et al. 2004). The created consolidated view is virtual, assembled dynamically and is often read-only. In practice, the registry index stores the global ID, links to data in source systems and possible data transformations. For these reasons, the registry style MDM is used mainly for real-time central reference. (White et al. 2006a)

Hybrid MDM hub is a compromise between thin registry model and the heavier transaction hub. It provides a single model to manage the identifying master attributes of the data (Loshin 2010, p. 169). This solution can be described as a reconciliation engine which attempts to solve the attribute partitioning problem and the synchronization problem at the same time. It is a hybrid model between different data sources and its client systems. (Talburt & Zhou 2015)

MDM hub offers some tools and possibilities of a centralized data management without the strictness of making every transaction move through it. In this architectural style, the master data is gathered from different systems and harmonized in a separate hub, thus it does not require seamless integration of the various source systems. There are two major benefits in this approach: overcoming the latency issues of a registry style hub and the heavy transaction hubs requirements of integration**.** A hub can be a justified architecture for MDM, in case it is important that the source systems continue on their own without a constant connection to the MDM system. MDM hub is an easy to approach solution in today's multivendor and multiproduct environments, where tight coupling of different information systems is not practical (Allen & Cervo 2015).

Nevertheless, there are downsides to MDM hub solution as well. Creating the hub requires copying attributes and creating a duplicate data, which increase the risk of inconsistency. The implementation is critical in the hub solutions, after all MDM is about solving the problems of inconsistency and not creating more of them. (Allen & Cervo 2015) The hybrid hub might turn out to be just another source for data, which needs to be governed and updated.

Transaction hub (Loshin 2010, p. 168) is a single centralized repository used to manage all aspects of master data. It represents the heavier style of master data management, where every master data item is created and maintained in the central repository. (Loshin 2010, p. 168)

In transactional hub approach, the data is created and maintained at a single centralized location, which prevents inconsistent data (Allen & Cervo 2015). All linked applications have the same global master data (Loser et al. 2004), resulting in the benefit, that data synchronization becomes obsolete, since there are no multiple copies of the same data (Allen & Cervo 2015). This approach relies on prevention and minimizes the need for cleansing and consolidation. Data stewardship and data governance procedures are also a lot simpler in this architectural style, as the roles and responsibilities are also centralized. The downside of this approach is that it is highly intrusive, expensive and time consuming (Allen & Cervo 2015).

A consolidated ERP system can represent a transactional system. It is usual, that the adoption of an ERP system generally requires that organization adopt standardized business processes, that are reflected in the design of the software (Hu & Morton 2008). ERP and MDM both aim at integration, but ERP follows a business process centric approach while MDM follows a broader approach to provide the unified data and distribute it across systems (Maedche 2010). All related applications must be modified to use the master data from the hub and the distribution of master data must be solved through an enterprise service bus (ESB) or service-oriented-architecture (SOA). It might be even impossible to implement, if the receiving applications are not flexible for customization to consume the data from the hub. Additionally, a major downside of this style is that it is difficult to create a data model which takes in account the needs of all business areas. (Allen & Cervo 2015)

3.5 Factors affecting the choice of a MDM architecture

First and the most influencing factor is the cost: consolidation and replication costs should naturally be less than resulting business benefits (Khairi & Shaykhian 2014). The total cost of one master record data is estimated at 3–5 USD (Nedumov et al. 2015). The cost might be difficult to calculate because the hidden costs due to bad data quality are not usually accounted for, and on the other hand the cost may seem high if not considering the fact that a good MDM solution has many indirect benefits (Bhansali 2013).

The costs increase due to acquirement of technical solutions but also due to the data governance related to the chosen MDM solution, which increases managerial and supervisory time (Bhansali 2013). Despite software vendor claims, MDM is mostly about defining organizational governance policies, not necessarily acquiring new information systems (White et al. 2006b). Vilminko-Heikkinen et al. (2016) note that the MDM tasks should be assigned to the person who is competent, but the lowest possible in hierarchy. Besides choosing the appropriate architecture, this approach to designing the roles and responsibilities might prove cost-efficient.

Architectural style	Data creation	Data maintenance	Availability of data	Data utilization	Amount of common attributes
Analytical	Decentralized	Decentralized in source systems. Federated ETL process.	Updated in batches Not guaranteed to be up to date	Reporting, analysis, central reference	Common classifications
Registry	Decentralized	List of global keys centralized, links to data in source systems	Changes available immediately with access	Real-time central reference	Global key
Hybrid hub	Centralized or decentralized	Global keys and attributes	Asynchronous with delay (batch) or real time and synchronous (inline)	Harmonization across databases for central reference	Global key and key master data attributes
Transactional	Fully centralized	Fully centralized	Directly available through service operated architecture (SOA) Receiving systems modified	System of record to support transactional activity	All attributes unified after parsing, cleaning and standardization

Table 3.1. *Architectural styles and their implications to data functions (Loser et al 2004; Radcliffe et al 2006; Loshin 2010)*

Starting MDM with a federated hub can be a safer way to proceed towards highly integrated transactional architecture once the stability of MDM reached (Loshin 2010;

Godinez et al. 2010). In some situations, copying the source data might not be allowed due to security, licensing or, for example compliance to laws regarding personal information. In these situations, a federated model can also be more justified. Although, in case there exists multiple source systems, a federated model quickly grows in complexity and might result in an unpredictable system. From this perspective, a centralized model delivers more stable results and is more predictable. (Khairi & Shaykhian 2014)

The speed of Master Data distribution affects the choice of architecture. Usually a bigger amount of data implies centralized architecture, while smaller sets of data can be federated quickly to deliver solutions. The choice of architecture affects the amount of data to be processed and queried, consequently affecting how quickly business benefits can be delivered. (Khairi & Shaykhian 2014) A centralized architecture can reduce costs and increase operational efficiencies related to MDM, but a federated model, where the scope and ROI (return on investment) is limited, the analytical results are quickly delivered is encouraged as a starting point (Loshin 2010, p. 11).

There are also technical and operational accessibility limitations factors to consider. A hybrid hub model compensates ETL query limitations by processing parts of a query already at the source server and thus, implies an integration and aggregation runtime. In the hybrid hub model, the easy access for queries implies higher customization and custom coding for different applications, and thus higher risk for failure. The centralized architecture has a centralized access and security which might be easier to manage. Nevertheless, the centralized model includes the risk of a system failure, and a backup system might be needed. From the business application point of view, the use of a centralized architecture might require locking master data, which on other hand might not respond to changing business needs quickly enough. (Khairi & Shaykhian 2014)

The data quality is also greatly affected by the choice of architecture. A federated model depends on data rules, compliance to them and the resulting data quality of the source systems. A centralized model is the one source which all applications depend on. Thus, centralized model has usually the ability to clean, validate and match the data to maintain the quality. Even though a centralized architecture creates dependencies, it is essentially non-intrusive to all existing applications. It can be seen as essential when the amount of redundant data increases substantially. (Khairi & Shaykhian 2014)

3.6 Creation of master data objects in different architectures

In any MDM environment, the creation of new data objects should be aligned with business objectives (Talburt & Zhou 2015). However, the different data source and destination systems, and their respective business units might have different needs to address, which make MDM a complicated issue (Dreibelbis et al. 2008). It is usual that every business unit usually considers its own data as the most correct, but in reality, it addresses only the needs of business unit (Loshin 2010). In distributed ERP systems, users might regard data quality in the whole ERP environment insufficient but are satisfied with their local ERP systems (Knolmayer & Röthlin 2006). The architectural choices have implications for the governance methods (Loshin 2010) resulting in variety of possible processes and workflows which to consider.

Different MDM architectures elaborated in chapters 4.1-4.4 imply different practices by which the new unique entities are created in the organization. In the analytical MDM, the creation of an entity is done completely to the source systems and possibly governed by some or very little common governance policies. In the registry style, the creation is also left to source systems and linked afterwards with other source system entities according to a centrally updated list. In the centralized hub models, the creation of the new entity can be totally centralized, or possibly implemented in the fashion of leading source system model, such as one presented by Loser et al. (2004).

Different applications might enable convenient ways to ensure that each point of decentralized data creation is compliant with the master data rules. For example, in Oracle's centralized MDM solution, the new items are input through a centralized web-based UI, where it is checked for redundancy, compliance to data rules, accuracy and completeness before actual validation and transfer to receiving systems. Nevertheless, in case of a centralized architecture, it might be necessary that the local employee makes a request that a new master data entity is created in the centralized repository. This can create delay for the local employee to actually get to utilize the master data.

It is important to decide whether there is a global primary key, which is utilized in all systems, or is there a different local key for each application. Global primary keys can function as a basis for entity resolution in the matching and consolidation processes but mapping corresponding entities can also represent a valid solution (Loser et al. 2004). If global keys are used, it can be implemented in two ways: letting the source system to bring their own identifier or doing it once and done method by utilizing a centralized look-up system. Letting source systems input their own identifier can function in a closed information system where data entry and exit points are controlled well and the data life cycle is long. However, it is not a practical approach for product or component data, which do not have a unique global code in the source data. (Talburt & Zhou)

When creating new master data objects, it is important to ensure the uniqueness of the identifier, so that additional duplicates are not created. Similar but slightly variant representations may have been introduced to the system, resulting in additional duplicates. Also, there might be a faulty perception that a record does not exist yet, when it actually really does. In this case, the variation prevents the identification of a specific match. Similarities can also be solved by probabilistic matching algorithms, although they might be time consuming and computationally intensive. (Loshin 2010, pp.187–188)

In the once and done approach, the global key is created centrally or assisted by a look-up system, which assigns the new entity a new and unique identifier (Talburt & Zhou 2015). The master data criteria should contain the common master data model, global master data key and related master data attributes, which are treated according to the data rules (Loshin 2010). In addition to the global keys, it should also be defined which attributes are constituted as global, and are thus created in the MDM systems, and which attributes are created locally (Loser et al. 2004).

Committing to master data criteria and demonstrating commitment is problematic and should be overcome by demonstrating how the data affects mutual processes (Vilminko-Heikkinen et al. 2016). It is not enough that there are common data rules and handbooks in place, but the information aggregated through master data efforts need to be available to the local processes (Knolmayer & Röthlin 2006). Defining the common master data attributes between business units is thus a consensus building effort in which the interests of the organization as a whole and of the business unit owning the data need to be taken into account and agreed upon (Loshin 2010, p. 74–75). In case there are conflicts, the data governance board should resolve the political issues regarding the required data attributes according to the organizations business objectives (Dreibelbis et al. 2008).

3.7 Distribution of master data in different architectures

When the new entity is created as a master data object, it is followed by a validation to central master data system or possibly a distribution into local systems. In addition to defining the local attributes mentioned by Loser et al. (2004), it is also important that a defined set of attributes is inherited from the global master when the master data object is localized (Dreibelbis et al. 2008). Some master data attributes might also have local needs, when translated from master data repository. In this phase, the quality of data is critical and different metadata models might increase the complexity of loading scripts. (Dreibelbis et al. 2008)

The consistency and availability are affected greatly by the architectural style because of the associated distribution method. In the reference method, there is no distribution of data, but only references, thus changes are available immediately with each access for the connected system. (Loser et al. 2004) On the other hand, it is required that this type of reference is up to date, which might create additional problems in availability.

The central master data system delivers the global master data attributes and additional attributes are added in the receiving systems. In a leading system the defined master data attributes which are created in the first system are copied selectively into the receiving system and additional attributes can be added locally. In case the data is created in a distributed manner or not distributed from a leading system, only the global master data key is distributed (in the reference system) or in the thinnest solution, only the standards are distributed. (Loser et al. 2004)

The process by which the data is created in the receiving systems when master data is distributed from and to multiple sources is of great significance. The data creation can be done in batch or inline processes. The batch process involves a bigger set of source data objects that are migrated into the master repository. In the batch processing data is collected from the sources, imported into a single location, and combined through parsing, standardization and matching. In the inline approach the data is embedded into operational processes when the new information is brought into the source system. The new data is then parsed and standardized in preparation for immediate comparison against master data registry and necessary corrections are applied when the instance is matched or created as a new representation. (Loshin 2010, p. 196) This approach applies even more broadly to consolidation strategies, introduced in the next chapter.

3.8 Entity resolution and consolidation strategies

Entity resolution is a process of sorting out information from multiple sources and deciding if the information refers to the same real-world entity. Other terms to call this process are record linking, data matching, or de-duplication. It is based on the global keys or some common master data attributes, which define the search terms and similarity between data objects. (Talburt and Zhou 2015) There are different matching tools and various strategies which can be used to parse, standardize, transform and ultimately match records (Loshin 2010, p. 184).

Parsing and standardization are the processes of recognizing patterns and value segments from the data values and feeding back the classifications into standard representation. These methods can be used to identify the patterns which can be utilized in actual entity resolution rules, rules that identify invalid and valid data as well as duplicates. Standardization is a special set of data transformations, which are applied to recognize and edit errors into acceptable formats. It is the process of mapping source data into a target structural representation. Slight variations in the data might cause confusion or ambiguity. In an analytical environment, where different data sources are introduced, the analysts need to recognize the supplied formats and create a canonized format in preparation for consolidation. (Loshin 2010, p.184–185)

Master data systems might need classifications to provide applications blocking mechanisms that naturally limit search space when trying to locate a match for specific entity. For example, product master databases may be limited when relying on specific words that name or describe the object. Additional classifications can be utilized to make recognition of similarity between entities easier and effectively organize master data objects to defined taxonomies. The matching and linking can be based only on text tokens such as product codes, but it is not as effective to organize master data. The classification can provide the entity resolution more refined content and context awareness. (Loshin 2010, p. 191).

When migrating huge amounts of data from multiple legacy systems, the existing data needs to be cleansed, so that the new MDM system and its aggregate data can gain the required trust in the organization (Knolmayer & Röthlin 2006). In case all of the data is cleansed immediately, when it is brought into the environment, it might result in additional work. In the business context, not all data is used immediately or all the time. On the other hand, cleansing the data that is required by business process on-demand might complicate managing multiple source instances. The hybrid solution between these is to determine standard form of the master data but also maintain the original source data. This approach creates additional duplicity, but the data is linked, so that additional knowledge considering the instance, as well as, historical traceability can be accessed. (Loshin 2010, pp. 194)

The data matching processes can be classified into attribute-based and record-based strategies. Attribute-based strategy is sorting out the different attributes in different sources, treating them individually and comprising a new record from the different attributes into a new record in the MDM system. Record-based strategy comprises all of the source record entities and attributes under one instance. These strategies can be further improved by introducing duplicate record or duplicate attribute removal algorithms. (Talburt & Zhou, 2015) The most important data quality dimensions that cause errors in data matching are accuracy and consistency, therefore pre-processing might be required (Christen 2012).

A variety of errors can surface when integrating source systems into one master data. Matching should not be never left to merely an automatized process, such as statistical matching, although it can make narrowing data easier. The errors can be divided into false positives and false negatives. A false positive occurs when two instances representing different real-life entities are incorrectly assumed the same. A false negative, on the other hand, occurs when two instances referring to the same real-life entity are not matched. Both integration errors represent risks to business operations. A false positive in product data might result in confused inventory management. A false negative, in turn, can cause missed opportunities when the product data is not analyzed efficiently. It is important to prepare for integration errors and devise impact assessment and resolution schemes for separating unique entities from the merged instance in case of false positive as well as deduplicating instances by resolving real-life entities in case of false negatives. (Loshin 2010, pp. 195–196)

History and lineage of the data used in consolidation is also of great importance when resolving integration errors. The full history of master data objects makes it easier to restore a previous state, in case false positives and negatives occur. The important history data attributes to consider are the changes made, date and time, rules applied in the merging, survivorship and de-duplications and the source system which affected the data. Resolving integration errors can cause additional problems in downstream applications, which also need to be “rolled back”. However, this roll-back might also be overreacting, if the actual business needs do not require full consistency of the data. (Loshin 2010, pp. 196–197)

3.9 Conclusive theoretical framework

The conclusive MDM framework of this research covers needs for master data, roles and responsibilities as data and architectures. The key points that are included in the theoretical framework are summarized in tables 3.2–3.5. These points elaborate the fact that a comprehensive MDM development spans from the top enterprise needs and attitudes all the way to the roles and processes, and to the underlying architecture and master data definitions. Furthermore, MDM maturity levels and different multi-business contexts bring more contingencies to consider.

Table 3.2. summarizes the key points regarding theme needs for improvement. This theme covers questions such as what are organizational attitudes toward master data, how it is defined and recognized, and managed effectively, what components it includes and how it evolves further.

Chapter	Key points
2.1 Data as an asset	Data should be considered as an organization-wide asset Management is needed to make data available Different kinds of data need to be managed differently
2.2 Identifying master data	Master data is can be identified as the most critical business objects that are utilized across many business areas, processes and systems. Master data is usually managed separately from other types of data.
2.3 Managing master data	MDM requires discipline and control over policies, responsibilities, data standards and systems. MDM aims at creating consistent and common data sets in situations where data is duplicated, fragmented or inconsistent. MDM efforts have many impacts on business on analytical as well as operational levels. MDM should serve business goals.
2.4 MDM components	MDM has multiple components that span from business processes management to architecture. MDM development can be approached in top-down or bottom-up manner, corresponding with process or data-driven approaches. MDM strategies can be further divided to problem and solution oriented. Identifying the needs and objectives is essential.
2.5 MDM Maturity	MDM should be seen as evolutionary, not one-off solution. Maturity can be classified to initial, reactive, managed, proactive and strategic levels. MDM maturity levels differ mostly in capability to utilize data across the organization and the attitudes and processes that support data maintenance.

Table 3.2. *Key points regarding theme needs for improvement.*

Table 3.3 summarized the key points regarding theme roles and responsibilities. It covers important questions such as how to structure and organize data governance to support MDM efforts, why the stewards are especial in this framework and which contingencies should be considered for close alignment of MDM and business goals.

Chapter	Key points
2.6 Data governance as an enabler of MDM	Data governance defines the decision rights and accountabilities for enterprise information processes. Data governance helps MDM efforts by defining practices, ensuring the monitoring of information policies and accountabilities. Data governance aims at preventing faulty data in the first place. Data governance can be implemented as an underlying consensus-building framework across multiple business units instead of centralized MDM which might cause political issues.
2.7 Data governance framework	Master data governance includes roles, responsibilities, decision areas and activities. Master data needs owner and maintainer roles. Master data governance roles can be for example: data governance director, oversight board, coordination council and data stewards. Decision-making roles can be further classified with RACI assignment matrix.
2.8 Data stewardship role	Data stewards are situated in between IT and business and need the support and resources of both. Data stewards collect, collate and evaluate issues, problems with data, usually in a specific business area. Positioning data stewards between owners and users of master data is essential. In different architectural choices, data stewards are usually positioned in between the source and target systems.
2.9 Factors affecting data governance style in a multi-business organization	Data governance and business strategies should be aligned. Contingent factors that can reinforce, conflict or dominate each other should be taken in consideration. Contingent factors regarding data governance include performance strategy, firm size, diversification breadth, organization structure, competitive strategy, degree of process harmonization, degree of market regulation and decision-making culture.

Table 3.3. *Key points regarding theme roles and responsibilities.*

Table 3.4. summarizes the key points regarding theme architectural approaches. It begins with addressing the question how to clarify the current state of architecture and start designing an architecture that is aligned with the organizations business goals.

Chapter	Key points
3.1. Clarifying current data architecture	Assessing current information structure can be initiated with questions what data exists, how it is managed and utilized and how it supports existing application architecture. Inefficiencies and redundancies need to be analyzed, and new data quality standards need to be clarified. New data quality expectations need to be communicated. Data quality needs to be measured with appropriate metrics that are derived from business goals.
3.2. Design principles for MDM architecture	Architectural design that is aligned with the organization business goals is the foundation of effective and efficient master data management. MDM architecture describes ways in which data is managed in logical and physical structure. MDM architecture should be customized by business needs, it usually cannot be readily purchased from software vendor as a ready solution. MDM design principles include being aware of business context, ensuring quality and integrating seamlessly, among others. Design principles guide, which MDM style organization should choose in the degree of centralization and level of common attribute definitions.
3.3. Analytical MDM as a starting point	Analytical MDM is usually the starting point for MDM as it addresses clear needs to support decision-making processes and comply with regulations. Data is extracted, transformed and loaded from operational source systems to a data warehouse, where it is turned into analytics and business intelligence dashboards. Analytical MDM usually brings data quality problems very visible, if monitoring has not previously occurred.
3.4 Registry, hybrid hub and transactional MDM	Registry, hybrid hub and transactional MDM represent different MDM architecture styles that vary for example in degree of centralization and amount of common attribute definitions. Registry style MDM is a simple master data reference or global index which represents lowest number of common attributions. Hybrid hub is a solution between transaction hub and registry style hub, having a common model to identifying master data attributes. Transaction hub represents the broadest consolidation of master data attributes and a single source system to which all target systems are connected to.

Table 3.4. *Key points regarding theme architectural approaches.*

Table 3.5. continues to summarize the theme architectural approaches. The important questions that are addressed are what factors need to be considered when choosing the architecture, and how it consequently affects the different processes of creation and distribution.

Chapter	**Key points**
3.5. Factors affecting the choice of a MDM architecture	Costs and potential benefits influence the choice of MDM architecture most. The degree of decentralization of information systems affects all processes. Synchronization methods and data creation differ in architectural styles and affect the speed, availability, accessibility, and quality of the data. Initial level of master data maturity might imply starting with light decentralized architecture style. High amount of data and needs for high reliability might imply a centralized architecture.
3.6. Creation of master data objects in different architectures	Creation can be organized in decentralized or centralized manner, but it should be aligned with business objectives. In decentralized architecture styles it should be decided if global keys are used in source systems or not. If global keys are used in creation, they can be used to consolidate master data later. If global keys are not used in creation, the linkage should be done in other ways of mapping the correspondent data. The keys need to be unique in order to prevent false positives and negatives in consolidation. Different technical solutions for data input and consolidation afterwards can be utilized.
3.7. Distribution of master data in different architectures	Defining global and local attributes is important to decide which master data attributes are inherited to the target systems. Consistency and availability differ greatly in each architectural style. The creation of new data in the target systems can be done in batch or inline process.
3.8. Entity resolution and consolidation strategies	Sorting out information from multiple sources and matching information that refers to the same unique entity and removing duplicates is important when moving large amounts of data from legacy systems into a master data environment. Data needs to be cleansed before transitioning to a new system so that an organization can gain trust to new data and systems. False positives and negatives can happen during entity matching which calls for rigid processes and systems that allow resolving the errors. Current data should be parsed and standardized according to the patterns that imply duplicates, valid and invalid data.

Table 3.5. *(Continuing) Key points regarding theme architectural approaches.*

As this conclusive framework reiterates: MDM is not just merely an IT problem. It starts with the attitudes and setting goals and needs to answer the important questions of how to organize around those goals and how to monitor and maintain the data quality. MDM has still a lot to do with IT. The architecture which links various systems which in turn contain the actual data are important to enable these processes and eventually, achieving the goals that are set. This framework formed as a thematic structure to form the questions in the appendix 1. As it can be seen, the it approaches the question of MDM from top-down manner starting with needs and ending in the architectural and data solutions.

4 CASE STUDY

4.1 Organizational context

The case organization consists of a parenting company which functions as a managing and consulting company, and two subsidiary manufacturing companies. All of the parts function in different regions of Finland, geographically a significant distance from each other. The enterprise had grown organically through exports and through acquisitions of small and medium sized manufacturing companies functioning in multiple industries. Out of these dozen subsidiaries, three were chosen for this research because they shared the same industry. Out of the three, two were researched. The organizational structure as well as the information structure was distributed and siloed, despite of common the efforts of the parenting enterprise.

The scope of this research in the organizational context is to focus specifically on the two subsidiaries which functioned in the same industry and shared similar ERP systems, and somewhat similar needs. Furthermore, the scope of master data was limited to the data about procured components. This data represented the most critical data, because of its high value and highly varied use in development, procurement, production and after-sales functions, as well as in enterprise-level reporting.

The driving needs for MDM solution were initially recognized on the enterprise level. The efforts to consolidate master data across these companies represents efficient reporting, removing duplicates in information as well as resources, and improving procurement efficiency. The subsidiaries were somewhat competitive among each other, but they are differentiated in customer segments and distribution channels. Thus, it was assumed that cooperation and consolidation of data objects was possible. Consolidating the data from different companies presents major challenges for the organization, while both operational and analytical uses and both enterprise as well as subsidiary level usage of the data needed to be considered.

The case company represents a case of a company which has grown through acquisitions and consequently experienced huge data management challenges. Thus, the motivation for this research is to find out how the organization can guarantee the appropriate quality of master data quality in future.

4.2 Methods

As mentioned in the 1st chapter, a case study method was chosen for this thesis. Case study method aims to understand the dynamics of the topic, mainly the interactions between the subject of the case and its context (Saunders et al. 2016, p. 184). The subject of this case study is the organization, and its embedded subsidiaries and furthermore the different stakeholders that created or utilized the master data objects. These roles were chosen in accordance with the different actions performed on data as well as who were known to have previously had a central role in with the data. The detailed list of subjects and respective roles has been given in the table 5.1. below.

Organization code	Interview date	ID
Enterprise	7.12.2016	A1
		A2
		A3
Subsidiary 1	29.11.2016	B1
		B2
		B3
		B4
Subsidiary 2	1.12.2016	C1
		C2
		C3
		C4
		C5

Table 4.1. *The sample organizations, date of interview and participant identifier.*

The research was conducted as an internal researcher, which granted direct access with regards to organizations key personnel and resources. The internal research approach provides significant benefits such as the thorough understanding of the problem, but it has a drawback, that the researcher should be very conscious about the assumptions that the position carries with it (Saunders et al. 2016, p. 208). Additional, the research was conducted as an independent researcher, not taking part to the organizations daily functions. This allowed for necessary detachment from the research problem.

The primary data collection method was an interview directed to the participants in the table 5.1. In total, 12 face-to-face interviews, each about an hour were held. Interviews were semi-structured. The questionnaire (Appendix A) was used to guide the interview and to make sure all themes were covered with every interviewee. Other questions were also asked during data collection, to gain deep understanding, characteristic to case study method (Yin 2008). Secondary data included the data and data models residing subsidiaries ERP systems, the collection of data handbooks and the analytical models of the parent organization.

4.2.1 Data collection

The research started with a selected literature review on master data management, master data architectures and master data governance. Scientific publications and books were searched from Google Scholar, Scopus and Andor. The search terms for master data management included the general terms master data management, MDM, master data, master data development and extended search terms such MDM barriers and enablers. For master data architecture and master data governance, the search terms included a broader selection. It was recognized early on, that MDM is a quite new research area and many ideas are still being translated from research learnings in data quality management and IT-business alignment literature. MDM architecture was searched with terms master data architecture, master data consolidation and entity resolution. The source material was not easy to find because the lack of literature focused specifically on master data, not general data management. The aim of the research was not to do a thorough literature review, but to provide adequate context for creating the research framework that would suit the case situation.

The data collection method for the empirical part was semi-structured interview. As described in chapter 1.5, the research purpose is exploratory. The semi-structured interview is less frequently used in this research purpose, while the unstructured interview is more frequent (Saunders et al. p. 392). This fits with the qualitative perspective that was chosen in order to gain deeper understanding to the research problem. A more unstructured approach was seen to be important also because it was known that the different interviewees had substantially different level of expertise on the subject. The unstructured approach was seen to be necessary in order to gain insight into all stakeholders needs. Also, the number of questions to be answered was broad, and the questions themselves quite specific and open ended.

In general, research interviews aim at asking concise and unambiguous questions, to which the interviewee is willing to respond and to listen attentively (Saunders et al. 2016, p. 388). Although the interviewees part-take to data creation, maintenance and utilization activities, the specialized terms such as master data management were not familiar to them. This possible lack of expertise also called for semi-structured interview. The social interaction plays more significant role, when the interview is not structured (Saunders et al. 2016, p. 391). This allowed the semi-structured interview to adapt according to the organizational context and the subjects level of expertise, and inquire into the problems at hand more thematically, guiding the interview with more specific sub-questions. Furthermore, the interviews were recorded and transcribed afterwards in order to recognize themes and the most relevant answers.

4.2.2 Data preparation and analysis

The collected data was analyzed in order to answer to the sub-questions 2 and 3; to define the current state of master data management and the possible barriers and possibilities to develop MDM.

The research recognized 5 themes from literature that were used as an underlying structure for the interviews to explore barriers and possibilities in master data management:

- Master data creation

- Master data maintenance
- Master data sharing
- Master data roles and responsibilities
- Master data architecture approaches

In addition to these recognized themes, it was important to take into consideration the needs and objectives for master data management. These themes were also used to code the actual interviews. These themes contain some barriers and enablers for MDM, which are further elaborated through the literature in the chapters 2-4 of this research. The conceptual framework was derived from literature. Master data management is a complex process with multiple viewpoints and steps. Thus, it was seen important to take into consideration the organizational as well as the technical architectural side of MDM.

In addition, the interview data was coded by the following framework in figure 4.1. This thematic framework was derived by reflecting the framework in chapter 3.9 with the actual interview data.

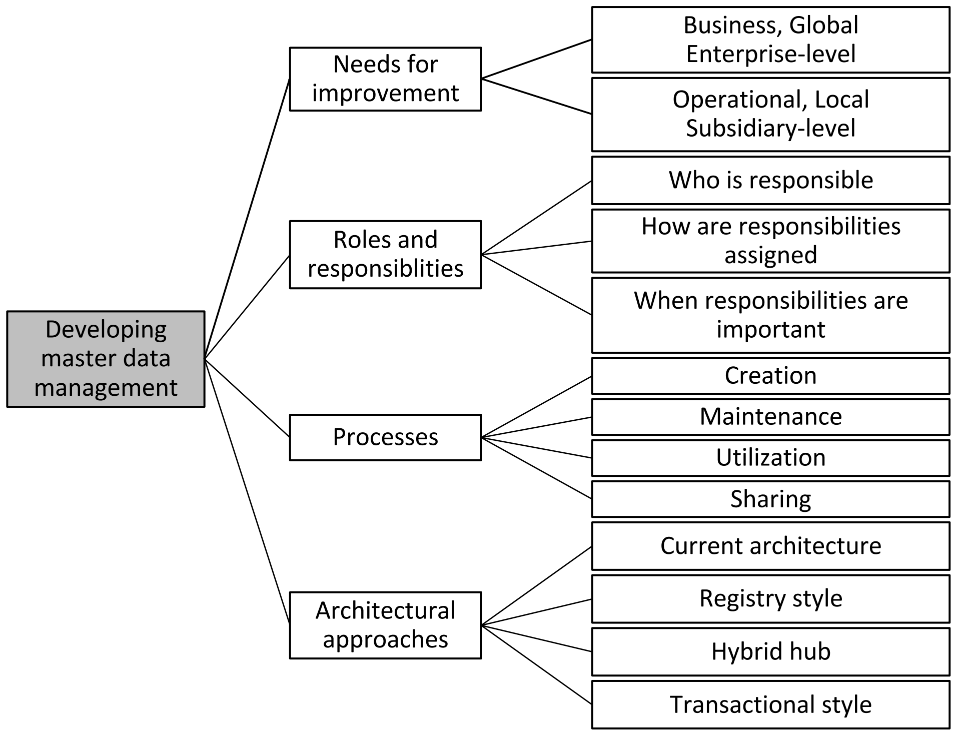


Figure 4.1. *Thematical MDM framework used in coding of the interview data.*

These points in the figure 4.1. were seen as the most important when considering the case organizations situation at the initial level of master data maturity. Furthermore, this framework was aimed for the case organizations specific context of having enterprise and subsidiary levels, rather simplistic current enterprise architecture and a low line of business IT-knowledge.

4.3 Conducting the study

It was recognized during the interviews that some of the questions might feel intrusive to interviewees. According to Saunders et al. (2016, p. 397) interviewees might be willing to participate but still be unwilling to discuss an aspect of a topic that the interviewer wishes to explore, because this would lead to information they do not wish or are not empowered to discuss, thus leading to impartial picture of the situation. The result of this response bias might be that the interviewee is cast into a socially desirable role.

The study was conducted at a respective subsidiary or organization as a face to face and one to one interviews. The face to face interviews were recorded and transcribed later into the thematic framework similar to the questionnaire form. The transcribed notes were then analyzed in terms of relevant themes and translated into English for the research. During the analysis phase, the themes were further developed and combined by creating flowcharts about the data processes and data flows through different organizational functions.

4.4 Factors affecting the results

The interviews were done in semi-structured way, to allow free flow of conversations. This allowed interviews to get deeper into the actual needs and problems that the participants had experienced. On the other hand, this turned out to be a heavy method afterwards. This method turned out to require a lot of analysis.

The participants represented relatively homogenous group of people, consisting of business function representatives. These participants did not have extensive knowledge about master data management nor IT subjects, other than what touched their daily work routines. This resulted in a narrowing the focus the research to concrete master data needs and processes.

Initially, the research was scoped to three subsidiaries. The third subsidiary was the biggest represented the best situation in terms line IT knowledge as well as data quality. This aspect might have affected especially the research's managerial recommendations, as the one company is also a major factor in considering the MDM for these similar subsidiaries.

5 CURRENT SITUATION IN MASTER DATA MANAGEMENT

5.1 Needs and objectives for MDM

5.1.1 Needs and objectives at the enterprise level

"The main point of this is to have the developers use more same components and same codes when developing new products" Participant A1

The main objective for master data management was consolidating disparate data across subsidiaries. Furthermore, the indirect objectives related to master data consolidation were to make common development, procurement, and warehouse management more unified and efficient. Starting to use the same components in a standardized manner, automating reporting were also seen important on the enterprise level. Other needs were expressed more in terms of wishes, than demands. The solution that is eventually chosen should not cause too much change resistance and should provide a straightforward consolidation process between the organization's subsidiaries.

The master data management efforts were seen to eventually go hand in hand with the actual standardization of the procured components. Therefore, a substantial part of the problem was also seen as one stemming from disparate development and procurement efforts. This problem had been readily addressed by regular quarterly procurement and development meetings where decision-makers from enterprise and the subsidiaries discussed also product and component-related issues. Nevertheless, these meetings had not addressed specific master data quality problems but had standardized some component through other efforts. MDM had remained a persistent problem to solve.

Common enterprise identifiers were seen as a method to drive standardization in development through effective sharing of knowledge, allowing the consumption to be measured accurately across subsidiaries and thus making larger purchases possible. This in turn was seen to make the production and warehousing more efficient and also allow distribution of production across subsidiaries. Furthermore, these benefits were exemplified by A1. Seeing easily which subsidiary has warehouse inventory on a specific component and seeing the price which other subsidiaries have acquired the part would result in great synergies between the subsidiaries.

The need for efficient reporting was clearly an enterprise-level need. In case the group names (classifications) for different procured components should be typed uniformly across different subsidiaries to allow the reporting of these items be efficient and possibly automated. It is usual to report for example the different component groups sales. There was concern about if this approach would work. The enterprise classifications were previously tried and failed, because the classifications had not benefited the subsidiaries operations in any foreseeable way, thus resulting in lacking motivation.

"The requirements for efficient use of data are common classifications and similar logic of belonging to a class across subsidiaries" Participant A2

At the enterprise level, there was also a clear need for an MDM solution that would be low effort and would not cause too much change resistance. Force and control approach to implementing the classifications to the subsidiaries had failed previously, and this was not seen as an effective approach. The required change from old to new habits could be too drastic and would only raise resistance and eventually lead back to the old habits of creating new data elements.

Finding similar components from another subsidiary was recognized as a central need. This could be achieved by many ways and one solution could be a common enterprise identifier. However, recognizing similar components could be easily achieved by a simple phone call inquiry to another subsidiary. In conclusion, participant A2 stated that there is no substantial benefit compared the efforts that the consolidation of even simple classifications would require. The common classifications would be a requirement for unifying chosen components in the development phase.

In conclusion, the needs for master data management at the enterprise level were expressed in multiple different domains. A general conclusion was that there was an evident need for MDM. From architectural point of view, the solution should be straightforward to implement and should support the business processes. The solution should bring straightforward ways to consolidate component data across subsidiaries. Organizational friction to change processes and take new responsibilities was seen as a clear barrier, and the benefits of consolidation were not that clear to all enterprise stakeholders.

5.1.2 Needs for MDM at the subsidiary level

The needs for MDM in subsidiaries were focused on operational needs and covered internal alignment between functions, more systematic data processes and agreed-upon documentation about data standards, naming conventions, adequate processes to maintain the data, and more extensive IT-knowledge. Different functions experienced MDM needs differently because of their different roles in the lifecycle of data, as well as, exposure and effect of data errors to their particular work.

The major difference between the subsidiaries was that subsidiary 2 had more employees than subsidiary 1. The organization was larger, and this allowed the data split into different domains of responsibility. Consequently, the needs were also split into differently according to the domains in subsidiary 2. However, there was more shared needs between the subsidiaries than unique ones.

A major need for internal alignment was stated in both subsidiaries. Conflicts in grouping and naming conventions between procurement, sales and production functions should be solved. These conflicts arose in situations, where the production function needed to put the component in a specific inventory group and sales function wanted to put the same components in the product-specific sub-group so that it fits their needs and they can find it later easier. Sometimes the new components were named by unique or ad-hoc standards which causes additional problems in finding the components.

There was also need for alignment in terms of subsidiaries' systems. Participant C1 noted that in his previous company, he used a transition file to move information directly from PDM to ERP. The consequence of this approach was that the information was synchronized, and any changes were made first to PDM design drawings and then transitioned into ERP. It had taken the complexity out of the update processes by providing a straightforward update method: all changes would be made just to the source system.

Finding parts easily was one of the most pressing needs in both subsidiaries. The components might be named differently even inside the subsidiary. It might take development over 20 minutes to find a specific component of a product. In spare part function, this is can be usually overcome faster by searching the part from the structure of the product you know it is a part of. If one does not know which product it belongs to, it must be asked from the production.

"An axe might be called splitting blade, blade, splitter, or any variation or combination of these" Participant B1

It was also complicated to find specific components by searching the common data warehouse with the business intelligence tool. For procurement, finding correct pricing information had led to discussions with suppliers with other subsidiaries procurers. There would be a need for common practices when creating the component data. Even a single page direction that states the different points to take in consideration when creating data could make difference.

In the spare parts function, the convenient search of spare parts was also seen as important to find after-care part and provide fast customer service. Standardizing names for components would be necessary for this function. There was also a need to standardize the names by putting first the general name of the component and the details after that. The number codes of the components were seen irrelevant, as long as the component names would make sense across the organization. Currently, the search results had to be so broad that it took a lot of time to resolve the right component from ERP systems. Because of the non-standard naming and classification practices, the easiest way to find a specific component was to ask it from the production.

The development function had a clear need to find also the testing reports easier. The components tested by other subsidiaries could not be effectively utilized. The current folder structure did not work, because there was a lack of common logic in how the parts are assigned to certain folders. Development had searched internet to find information about new components, which was not an easy task. Usually, the relevant information could be found in testing reports. Thus, it would be of great benefit to find internal documents about tested components. Also, the supplier's order number should also be found in the testing documents and ERP immediately, so that extra information could be searched of the components online.

The component test reports residing in the enterprise server are in a very odd folder structure which did not make sense for other people trying to find them. Some kind of word or tag-based search could be convenient to browse through the test reports and actually find something useful. Component groups should be clear and there should be drill down options from which to choose. He stated that the definitions require a lot of work, and now it is just easier and faster to ask from production or another subsidiary. The next statement depicted the situation clearly:

"It would be easier to get a full list of 1000 reports and search from them." Participant C1

In addition to seeing clearly which components are used inside the subsidiary, there was some need to recognize what components the other subsidiaries are using. This information was currently asked by phone or from the other subsidiary or the actual supplier of the procured components. Anyhow, this was seen as a problem, because one needs to specifically know what to ask, to find out other components. Removing the duplicates inside and across the company was deemed important, and to do this, there was seen a need to rename the master data categories with the same terms. The exchange of data between subsidiaries and resources should be defined clearly. Subsidiary 2 had already some data and structures copied directly from the third subsidiary (the subsidiary which was limited outside the scope of this research).

A major stakeholder in both subsidiaries considering the bought component master data was the procurement function. The component data was utilized mainly in the procurement processes. For the procurement function, the most important data to have were the manufacturers component number and the subsidiary's internal identifier for the order process. Later the procurement needed the data for checking the warehouse balances so that he could react by ordering more.

For the latter functions in the data lifecycle, such as sales and spare parts functions, it was also important to see from the product page which parts had been changed, when, from which part and to which part, as well as what was the products serial number that had specific parts. This information should go all the way to the level that it is changed in the spare parts book. There should be a clear picture from the perspective of after-marketing and spare parts function about which information is missing.

"The more the subsidiaries would have common data, the easier it would be to search new components across subsidiaries and initiate other types of co-operation" Participant B1

There had been previously errors in the component data due to loosely defined processes and humane errors. Sometimes the warehouse balance had been incorrect due to a humane typing error in the production phase. The production might have forgotten to mark that a product had been finished and would be extra components according to the system. When a component was replaced in a products production phase, the change was not updated to the structure of the manufactured product immediately. This had caused the inventory count to reduce the old component, and in equal quantity, add inventory count for the new component. Negative inventory counts were therefore usually checked with ERP reports and maintained after these incidents. Nevertheless, the approach was only reactive.

The reactive approach did not meet the sales functions needs either. Sales functions utilized data to calculate cost prices and when the component data was changed, it was not necessarily updated to the product structures, therefore resulting in wrong cost prices, and hindering accurate pricing. When a change in similar component is noticed, it is slowly updated into every location, making warehouse balances, prices and sales price estimates complicated.

From these points, it is also important to note the limitations of the ERP system, as it required update processes to multiple locations inside the ERP. There were also needs to edit the master data objects names and possibly remove the objects. There might happen a typing error and the product is already in use. Some component changes had taken years to notice that it had not been updated to the structure of the product. This also implied a lack of continual process for the checking and updating of the data. Also, this need to update the changes would also surface only when someone separately asks for it.

The needs for MDM in both subsidiaries spanned all operational functions and domains of MDM. Product development would need common naming conventions to find components from testing documents, as well as from the other subsidiaries ERP. Procurement and spare parts function would need to find the components from the ERP and to keep the warehouse inventory up to date. Production needed to gather the right components, and their work was heavily dependent on right inventory counts. For sales, the biggest need for MDM was to calculate cost of goods accurately. Also, the needs touched the systems and data architecture in each subsidiary. In conclusion, these needs for accurate data and master data were broad, and more leaned towards operational needs compared to the needs at the enterprise level.

5.2 Creation of master data

5.2.1 Creation of master data from the enterprise perspective

The process of creating a new master data element varied in each subsidiary. Data was not created at the enterprise-level, but the enterprise had some influence over the subsidiaries' processes. On the enterprise-level, the view was that a new data element should be created when a new component is introduced in the zero-series production batch of the product. It is up to the development function to add the new component when the product moves from prototype-phase into actual series production. In case the component is still in prototype or test phase, the creation of new master data object was not mandatory, and the component was grouped in the ERP under a miscellaneous category. All expensive components, despite small batches, should be created immediately.

Adding test components as a separate master data object might not make sense, as it takes too much namespace and merely complicates the inventory lists when the component is used only once. Subsidiary might receive sample components that do not necessitate data creation in the ERP system because they do not require billing. However, these sample components should be created in a testing form with the testing diary information: how the component performed, did any problems surface, and so forth. It was essential to also add serial numbers of the new products under test. Test reports represented an important early phase data entry point, which could be utilized by other subsidiaries as well.

When the components are asked from the supplier it usually means paying for it and this requires the component to be created in the ERP. The underlying requirement for creating these data elements is the compliance to financial reporting standards that require R&D components to be accounted for separately and involving a product.

In conclusion, the new component data was created only in subsidiaries. From the enterprise-perspective the subsidiaries created data in form of test documents during new product development, or to the ERP as a new data object when starting production phase. This view was somewhat aligned with the next chapters presenting the subsidiaries data creation. However, the enterprise-perspective was focused on testing and production more than product development and the PDM system that was utilized in subsidiaries.

5.2.2 Master data creation at the subsidiary level

The process of data creation in subsidiaries involved development and procurement functions, PDM and ERP systems as well as separate testing documents, spreadsheets and supplier information. The processes had been evolved by the operational needs of the subsidiary, such as the cost of goods calculations and procurement processes. There were some contradictions with the definition of a unique component between the subsidiary and suppliers, which seemed to have no clear solution. The major differences between subsidiaries' data creation were that the subsidiary 1 had decided to centralize the data creation for a single person because of problems in duplicate data and different practices between creators. Subsidiary 2 had multiple employees responsible for data creation, each responsible for creating data in their domain.

The process by which data was created was described in very linear way. First, the development designs and draws new machines and tests possible new parts. After all of the tests have been run, the product parts are listed in a spreadsheet for costs calculations and the product parts can be ordered. The product can be moved to zero series production, when the initial design is approved. All new procured components are created the first time to the ERP system, and the product structure which contains all parts and work phases is created. This was the fairly linear process of component data creation in both subsidiaries.

In the prototyping phase before manufacturing, it is common that there are changes are made on the basis of tests. Component testing takes a substantial amount of time. If the component meets the requirements, the company will begin to use it. There seemed to be a small contradiction between the views of the development and procurement, whether the components should be always ordered by procurement with a certain data element or not.

Already existing components that had an internal identifier in the cost of goods list are searched from ERP and also added to the product structure. Technical names and prices had been acquired from the supplier, and they were listed in the spreadsheet. The internal name and identifier were marked with a question mark for the new components in the spreadsheet. It was estimated that about 10 percentages of the parts are totally new, 80 percentages are old parts and the remaining 10 percentages are product-specifically customized parts. The customized parts need to be estimated separately for each product, thus creating huge amount of customized parts in the ERP.

At subsidiary 1, most of the component data were created by development and they had their own memorized practices and standards in both subsidiaries. The first moment component data were created was in the products' technical drawings, where the development give some components an internal identifier. In the subsidiary 1 the component was classified in a certain category with two initial numbers. Subsidiary 2 had their own way of classifying the components. There existed a data creation guide in subsidiary 2, but participants did not know if it was updated or not.

"I create data very rarely, and when I do, I need to check the guide to get it right" Participant C1

Participant B1 saw that there were problems with the way data creation started. There had been various manual steps to make it happen. First of all, the data element had been named internally in the drawing applications. Product data management (PDM) application had been acquired by the enterprise but there had not been adequate training in the subsidiary to start using the tools instead of the old way of working. The users also expressed worries that even though they could get support, the application consultant does not know how the application should be used specifically in the subsidiary context. Implementing the product data management tool would need practice in real situations. The subsidiary's development function did not have the time to start using the PDM properly immediately but was aiming to use it in the development of the next new products.

The data creation of purchased component is simple, because it did not have its own cost structure, unlike the components manufactured by the company. In the creation phase, the classifications are essential. Component category and inventory category need to be right so that the expenses and spare parts sales can be later reported correctly. There existed a common naming policy inside the subsidiary 2 for the self-made components, but it was specific enough for most participants. There were so many things to remember, that documentation was seen essential. There had existed a guide for creating procured components as well, but it had become a routine task for most people. Still, some things might have be missed because of over-reliance on routines instead of the common guides.

"Even just a simple paper guide on the wall could explain how you do a new data object and fill the fields correctly" Participant C1

Even though the procured component data creation was centralized for a single person, there had been problems naming the components correctly and finding them from the ERP in later stages. With the self-made components, this problem was also evident, because the names were usually ambiguous. As a slightly larger company than subsidiary 1, it was concluded that there needs to be multiple responsibilities to split the data creation into logical domains such as procured and self-made parts. However, there was still a clear need for data creation standards and documentation.

Subsidiary 2 had divided the responsibilities of data creation into different domains. One employee was responsible for self-manufactured components in the system, their structure, work phases and control methods. which was in some ways more complex than procured component data. Data about self-manufactured components was input to ERP in accordance with a self-made guide by participant C4.

The business intelligence solution was seen as an application that could be utilized more, especially in the procurement function. If there exists a similar procured component in another subsidiary, for example, searching the enterprise-wide data warehouse could be beneficial. The procurement understood the common classifications so that the products could be put in same classes in the BI solution. Nevertheless, these classifications were not utilized currently in the creation of new data. They were introduced in 2013 and the main reason that they were forgotten seemed to be personnel changes and inadequate documentation of the data creation process.

There seemed to be a contradiction between the definition of a unique component between the subsidiary and the component suppliers. The supplier might have changed their component identifier when some even a minor change is done for the component. This identifier can change only inside one sales area. Therefore, the three subsidiaries might be using the same component, but with different supplier codes. This had been tried to overcome by making supplier contracts at the enterprise level. In the end, it was seen very important that the subsidiaries would rely on their internal codes, not the suppliers. This was complemented with the process that even if one thing such as component color is different, a new component data row should be created.

The number of people who were able to create master data object had resulted in many problems. There were many unclear roles and because many different components categories without explicitly specified standards. Components starting with number 97 are electrical components and components with number 95 were spare components, for example. These numbers represented existing data standards in subsidiary 1 that were not documented. The documentation situation was similar in subsidiary 2. Every employee, who was involved with data had to remember, what the arbitrary numbers signify.

Fragmented roles and lack of documentation in data creation had resulted in these various classifications, which in turn, resulted in inconsistent data. The unclear responsibilities in the creation of data had previously resulted in a lot of duplicate component data. After many duplicates had been noticed, it was decided that the component data creation should be centralized for a one person. This seemed the only way to make the component data uniform.

It can be concluded from the interviews, that data creation was not the most pressing issue regards to MDM efforts in subsidiaries. This was mainly because of the anecdote that only about 10% of the parts are new and need to be created. Nevertheless, there were many points of development to clarify the data creation processes, architectures and roles so that they align better with the subsidiary goals. To initiate MDM, it is logical to start with aligning new data creation with the MDM goals and to enable the 10 percentages of parts which are new, to be created according to new data standards. Creating, documenting and communicating these data standards across subsidiaries seemed necessary to achieve uniform master data creation.

5.3 Maintenance of master data

5.3.1 Master data maintenance at the enterprise level

There was a common understanding at the enterprise level, that the current master data maintenance level was very low which resulted in low master data quality and many obvious things could be done to fix it. First of all, checking the data quality should be constant in the organization, meaning processes as well as a technical solution. There was a view that master data quality is everyone's responsibility. This, however, implied in subsidiaries that not a single person actually pays attention to the master data quality. The maintenance was seen as reactive rather than proactive and was limited to the subsidiary's scope.

The maintenance of master data was seen only to work if there would be a responsibility to follow guidance, which is closely connected with making daily operations function properly. The current approach to data maintenance was not based on enterprise data strategy, or anything even related. The main approach to maintenance was to keep daily operations running smoothly. It was seen evident at the enterprise level, that in order to raise the master data maintenance level, it should be similarly linked also with daily operations.

The maintenance operations that were previously done were mostly reactively and related to the subsidiaries daily operations. Occasionally, there had been requests that specific things should be marked by everyone. Nevertheless, these procedures were not systematic, it was in its initial phase that something like this is called for. An automatized, quantifiable checking should be implemented to make sure that the classification codes are typed correctly and the most obvious errors, for example, in warehouse inventories are corrected.

Another clear need for developing data maintenance were the inventory needs. Every major component should be kept up to date and not just on yearly basis. This would allow monitoring the inventory turnover and usage for components. Thus, they could save the namespace and remove components that were used only once. The components that are not part of any product structures should be subject to removal. This is where the limitations of the ERP played a big role. The component data could not be removed from the system after it has been once created. It could be only removed by the participant A3 on the enterprise level.

Some of the maintenance problems were seen stemming from understaffing problems. Also, the different subsidiaries have different organizational structures, making assigning the roles and responsibilities difficult at the enterprise level. The problem of common processes stems from available staff. Also, when the ERP systems were implemented in the subsidiaries, there was no clear centralized guidance or support to use the system. This had also resulted in a variety of different practices to maintain the data.

In most subsidiaries, there was only one employee who maintain the data in a certain domain. The procurement functions had a lot of work in all subsidiaries. The roles and responsibilities regarding the maintenance processes were very unclear. On the other hand, there could be seen a clear need to have the data maintenance processes in place so that the enterprise "would not sink into a deeper swamp" (Participant A1). It was stated that a common guidance to data maintenance should be made as soon as possible.

The problems in master data maintenance arose usually when another function discovered some inconsistency in the data. There existed some checking for quality and requests maintain and fix the data, but these were not well organized to be a core part of organizational processes. It was up to the awareness of a person to decide how and when to fix certain data elements. It was seen important that this kind of maintenance procedures should be clarified so that someone would make them.

It was seen important that the data maintenance operations should be considered on a continual basis. Cleansing the data from errors should be done on a daily basis so that the errors do not accumulate over time. Maintenance operations should be done at least when the inventories are count, once or twice a year. There might be some typing mistakes when the inventory is marked. 100 items in the system might in reality correspond with 1000 items in the warehouse. In the production phase, the inventory might go negative if an old component data is forgotten in the product structures. When the inventory of one component goes to negative, the new replacing components inventory is compensated with that much additional inventory. A simple mistake is followed by two incorrect inventory counts.

The data maintenance responsibilities should be clarified for someone, but the problem is that the organization cannot assign a person to do it full time. Usually the participant A3 is seen responsible of the data quality. Participant A3, having access and rights to all data in the enterprise and subsidiaries, thus making it easy to spot inconsistencies and errors in typing. Data inconsistencies have been tried to resolve afterwards by manual requests by phone or email.

There were no clearly marked person who would be responsible for the maintenance of the data. Each company and each function are fully responsible for the data they create and use. Inside subsidiaries, the maintenance responsibility was shared across different functions. Procurement is responsible for the procurement components, manufacturing is responsible of the self-made components and sales is responsible for the self-made components that are directly sold. The maintenance responsibilities included only the implicit task of keeping the daily operations running smoothly. There was a lack of responsibility to address enterprise level needs and a lack of processes towards proactive, long-term master data quality in the subsidiaries.

In conclusion, the obvious things that could be done to fix maintenance include automatic, programmatic checking of data quality, clear responsibilities, having enterprise-wide documented data standards and assigning enough time and resources for data maintenance tasks. These were seen as some steps from the current reactive data maintenance toward proactive data maintenance. However, it was seen that data maintenance should be prioritized for business needs, and not done just for the sake of having good quality data.

5.3.2 Data maintenance at the subsidiary level

It was important to constantly maintain master data from several perspectives. However, most maintenance processes in subsidiaries were reactive, not proactive. The barriers included lack of documentation and proactive maintenance processes, the same technical and usability issues with the ERP system and updating the components and product structures. However, the process that seemed to cause the most problems was a change in the product structure during the production phase. The responsibility of maintenance was unclear in the subsidiary 1, and the development and procurement meetings were seen as the only factor currently supporting proactive data maintenance. In subsidiary 2, the responsibilities of maintaining different data domains were assigned to different employees on the basis of function.

Supplier code for components might change and to order more of the component, the code needs to be maintained in the ERP. The supplier code might change even because of a small change in the component, but it needs to be updated. When a component is changed by the supplier, the supplier code is changed. The component data of the subsidiary is replaced from the product structure only if the part is fully replaceable by the new component in every attribute. Uniqueness or sameness of a component can be complicated to establish and necessitates the physical component in addition to rich data about the component attributes.

In case the product structure is updated, the new component does not change the currently open production orders. When the production order is made, it forms on the basis of the current product structure. In those cases, it was required to remove the production order and make a new order. The parts had to be changed separately for the production orders or alternatively deleted and re-ordered. Either way, a new replacing part during production represented a lot of maintenance work. This situation was problematic for both subsidiaries.

The most important things to maintain are the setting times, materials, duration of work phases and the costs of the product, aggregated from the component prices. Production data was followed on the daily basis, because supplying the parts to production was seen as very hectic process. Previously, there had been a lot of maintenance and attribute updating tasks after the creation, but the participant C2 stated that the employees had learned, what data they need, so that is was taken care of in the data creation phase. This kind of iteration of practices seemed to be more present in subsidiary 2.

The ERP system had technical limitations for effective maintenance of the data. Once the data object had been created it could not be changed afterward. This problem was stated broadly in both subsidiaries. It was overcome by changing the name of the data object and changing the state to out of use. In subsidiary 2, these components were seen as items that should be subject to removal, because of limited namespace for new components.

There were also other usability problems, that stemmed directly from the ERP's features or the lack of them.

Lack of support for historical metadata, especially concerning the product structures and components caused major problems in spare parts and after-marketing. This happens in case a new component data is created and replaced over the old component in the product structure. Participant B3 stated that it was practically impossible to find out what component was used in a product that was built 10 years ago. The ERP shows which of the components and the structure of a product were changed, but it was not clear, from which component how it was changed.

The maintenance status with product structures was recognized to be at very poor level in subsidiary 1. The employee who used to be responsible of the tasks had left the subsidiary, and no one had taken the responsibility of his tasks. The maintenance is usually done when while calculating at the cost of own production in the ERP. The ERP does not always aggregate the updated component prices all the way to the cost of complete product structure. It is also important to note, that the component data also expired. The same part can be used in many products and places of the product, which adds even more complexity for maintenance processes.

The general consensus around data maintenance in the subsidiaries was that the responsible should be the one who creates the data. However, it was not always clear who is responsible for creating the data elements and it was usually another person in the later lifecycle of the data, who had noticed the data quality problem. For example, in subsidiary 1 the new product structure was created by the development function, but the ERP structure and component data are created in the procurement. When a component was changed it might have been updated to the ERP, but it should be updated to the structure as well. There was no clear responsibilities or processes between the procurement and development for the maintenance and update processes. This had resulted in inventory errors for example.

The roles for maintenance were unclear for the component master data in subsidiary 2 as well. The responsibility was on either up to the development or the procurement, or in cooperation together. Most participants saw development function as responsible for updating the components to the product structures. The procurement, in turn, might make component changes when the supplier changes the component, but the information does not necessarily reach the development. The development might have also made changes to components and then the changes were not put all the way to ERP. When these changes occur, the component data should be updated to the right product structure immediately.

However, participant C3 stated that updating the structures for products is the productions responsibility. There was some responsibility divided by how the parts relate to the function. The bought components were clearly the procurements responsibility, but the structures were the procurements responsibility, thus affecting the cost structure of the product and inventories of bought components.

Some historical factors also seemed to affect the current state of data maintenance efforts. In 2013, there was an initiative to update enterprise codes for the components to make master data more readable for enterprise needs and for other subsidiaries. These fields had been used by the participant C2, but this process was eventually forgotten. It had been left to the subsidiary to take care of them, but it had not been an integral part of the daily operations. The enterprise had more use for the codes than the subsidiary procurement. Another historical fact was that there was still some legacy data remaining from the previous ERP system which the employees had tried to maintain.

In conclusion, the barriers to data maintenance were broad in subsidiaries. The barriers included technical limitations of the ERP, lack of proactive maintenance, lack of subsidiary integrations between subsidiaries, unclear responsibilities, and lack of documentation regarding core processes. One clear problem was defining the roles and responsibilities clearly across all data domains and the core processes of data maintenance including the replacing components, updating the component attributes, and updating product structures.

5.4 Sharing master data

5.4.1 Data sharing at the enterprise level

The current problem of master data sharing at the enterprise level could be seen in the difficulties of getting and combining the different consumption volumes from the three subsidiaries. This was seen to be caused by different strategic choices, lack of common data standards and a failure to implement enterprise classifications in the past. At the enterprise level, employees showcased will to have uniform guidelines for master data creation and sharing, if they can be made. Currently, it seemed impossible, because of lack of resources it is difficult to define common processes across subsidiaries.

Making operations more standardized and making development, production and replacement parts function efficiently were seen as some benefits that would follow from sharing a single version of the data between all actors. The component data created at developer level also lifted a problem. While on the other hand it was seen good to have original and inventive development, there seemed to be more need to make operations efficient by standardization. This would call the developers to share data as early as possible and not make custom solutions to every problem.

There had been an example of one component which had been standardized. This example epitomized the problem. When everyone else in the organization thought the component should be standard and easy for the end customer, the developers had made a variety of different custom-made oil tanks for each product. Balancing innovativeness with standardization seemed to be a very important strategic choice underlying this problem.

Some data had been shared among subsidiaries, and the enterprise had access to all of the data. Nevertheless, sharing data between subsidiaries was seen impossible because of the different creation methods. Subsidiaries could not read each other's ways of inputting the data. There existed so many variations to a product name that people do not know what to look for. So, the problem of sharing data traced back to the creation of the data and the naming conventions.

Another way to fix it would be to have a shared library of the most common names to use and use this library in daily processes. Nevertheless, it seemed a difficult task, even if only one employee is responsible for the naming. Changing the naming conventions also leads to the problem that the names should be changed everywhere from spare parts books to even the old product structures. Changing the naming conventions needs to be justified financially to make it actually reality.

One strategic choice affecting the data sharing was that the subsidiaries were seen somewhat competing against each other in the domestic markets. Therefore, a subsidiary might want to introduce product features faster than another subsidiary and these properties would be adopted by the other subsidiaries and competitors later. The competition used to be a lot worse in the past, when subsidiaries could intentionally get a different part to say to customers that their part was better in some way.

A top-down command and control approach for setting the general data creation guidelines was seen as something that would not work. On the other hand, defining the common classifications and starting with some bigger component group that was seen as a way to bring value. For example, all procurement data should be common between the subsidiaries, but starting the commonality with some specific components to demonstrate the benefit would seem practical. This might overcome the worry presented by participant A2, that the company had tried this common category approach and it did not help before. The problem with this approach was that every subsidiary wanted the changes to be implemented in their manner, and they would not want to adapt to others way of classifying components.

Not a single subsidiary had a "best in class" or explicitly planned approach to the data classification problem. Despite this, participant A2 stated the worry that the enterprise could not come up these common practices for the subsidiaries.

"The subsidiaries have practices that have evolved through their experience and we can not necessarily understand clearly enough at the enterprise level" Participant A2

However, considering the common enterprise classifications that had been presented to the subsidiaries had posed no threat to existing practices. It had just presented the subsidiaries with extra work, and no clear benefit for their daily tasks.

The enterprise classifications to share the data were not seen as something that would work. The number of different classifications was seen as unmanageable between the subsidiaries. The problem with previous enterprise classifications was that there was no clear instruction why and how these classifications would be used. After grouping the components into these classifications, it was seen important to actually look inside the category, see which components were the same and combine the different elements with each other. This kind of system was seen as the only way according to participant A1. Participant A2 stated that it would be impossible for the subsidiaries to make the internal codes with common practices because they are currently in very different form.

5.4.2 Data sharing at the subsidiary level

Data sharing benefits had been widely recognized in subsidiary 1. These included seeing what other subsidiaries were using, their availability, costs, past purchases and replaced parts. The channels to share information about components between other subsidiaries were by phone, email and face-to-face meetings. Currently, actual data sharing did not take place, although, each subsidiary had access to the business intelligence tool. Because of differences in naming and classifications, the data in the BI tool could not be utilized effectively. Data sharing in subsidiary 2 was in conclusion on somewhat better level than in subsidiary 1, which can be partly explained by the number of employees and resources.

Participants stated that the increasing data sharing has the potential to help the subsidiary with everything from better component availability, lower expenses, easier purchases, product maintenance and easier replacement of spare parts. The supplier holds the parts that are common for many subsidiaries even more in the warehouse, thus making it flexible and more secure to purchase the components. Thus, the development functions expressed a great interest in seeing the shared components straight away, and to get the information about what components other subsidiaries are using.

The development had problems in data sharing because the testing folders in a specific hierarchy, and the classifications were subsidiary specific. Finding certain components in the testing folders, such as valves, was described as a complicated process. The grouping was based on specific classifications, which were not the same in all subsidiaries. Thus, searching test reports from other company that did not work. A search function based on tags or common search terms was suggested as a way to make sharing the component data and testing reports easier.

It required a lot of additional information, to actually know if the part is compatible for other use cases as well. Sometimes, the common suppliers might share information that other subsidiaries inside the enterprise are using the same component. In fact, the contracts obligated the suppliers to give this information. Nevertheless, the information might also be also decentralized into a separate information system for each sales region of the supplier company. This makes supplier informing uncertain and places more responsibility for the procurement to ask about the different components from the other subsidiaries.

In procurement, it was seen important to share the pricing info and have discussions with other subsidiaries when a new product arrives. Currently, the procurement had asked other subsidiaries by phone and email if they use the same components. Nevertheless, many participants stated that when one specifically has to ask for information, it is not necessarily always asked. Many things were based on assumptions, and information was not shared when the employees were not aware enough about all the possibilities and situations, where discussion with another subsidiary would be meaningful.

Sharing should occur also in the domain of best practices, not in only the domain of data. There was recognized some obstacles for sharing best practices as well. One was the fact that there might be some conflict due to competitive strategies between the subsidiaries. The other fact was, that the products were essentially slightly different. The subsidiary 2 manufactured its own parts while the other subsidiaries ordered them from suppliers. Despite the situation, subsidiary 2 had made another subsidiary a product with their branding, for a market where the other subsidiary did have no presence yet. This can imply that even deep co-operation was seen possible, if it had clear benefits.

Data about tested components was shared in the common component test library, which was shared between the companies. However, the test documents resided inside folders and the classifications are not defined across subsidiaries, making it difficult to find relevant component tests. There were also product drawings inside these folder structures, but it seemed, that it is faster to ask this information over the telephone. There should be other ways to see the procured components. Procurers and developers had to ask the components, but the information should be available faster. The rich information about components was seen important before the procurement when trying to define if the component would actually fit a purpose.

Sharing the data inside the company was seen to work poorly because there was no way to find the needed data easily. This stemmed mainly from the lack of standardization of product names and classifications. Participants stated that he had to browse unnecessarily long lists before they could find the right part, open the PDM drawings, or in worst case do guesswork. The development first usually asked the production about who knows about the components. If they remember that a certain part is used in a product, it could be found in the ERP system from the product structures, not directly and conveniently with a keyword.

Participant C1 had previously worked in a company where he utilized a transition file that allowed exporting of the product information directly from the product data management tool to the ERP. He saw this process to reduce a lot of manual work. However, there had been some role and responsibility conflict in the process of informing development about product changes. Because of this, the product structure update tasks were transferred from development into production in subsidiary 2.

Sharing the component data and information was seen only as potentially important. If, for example, all valves would be tested in one subsidiary only, it might make the new products' time-to-market shorter. However, the test reporting was seen to be on an insufficient level inside the subsidiary and sharing this test information with others would require more focus and time. Some component data sharing had been facilitated between subsidiaries, but with a focus on information systems, and not the practical utilization of the information. For example, participants C1 and C4 had used the existing BI solution, but had not found it useful in daily situations, and therefore stopped using it.

The subsidiaries' procurers and developers met every two or three months and focused on coordinating common purchases for materials and components. This was very manual work and required a lot of resources to organize all important stakeholder across the country into a single place. These meetings were seen to work well for the development function to coordinate common development plans, although there were various topics discussed in the same meeting.

In conclusion, data sharing in subsidiaries included many barriers and points of possible development. The benefits of sharing the data were not clear to all participants. The most interesting point of sharing was in the product development and procurement phases. Nevertheless, the possibilities could not have been utilized without common data standards, classifications and clear responsibilities. Currently, the enterprise supplier contracts that obligated suppliers to share same up to date data to all subsidiaries were seen as an important factor to facilitate data sharing.

5.5 Roles and responsibilities in the organization

The roles and responsibilities concerning MDM were currently undefined in the enterprise as well as in the subsidiaries. There had been many changes in the subsidiaries personnel and roles during the past few years. On the enterprise level, the responsible for master data was seen to be the participant A3 at the enterprise-level. He had access to all subsidiaries data and who had implemented the downstream analytical solution. However, with some exceptions of data being also maintained, this solution represented mostly the analytical use of the data. Some policies concerning MDM had been clarified by participant A2, who had in turn access to supplier data, which was seen also as one way to recognize similar components and ultimately consolidate the master data.

The procurement and development functions in each subsidiary were seen the most important in terms of standardizing components and making the master data possible. There had been regular meetings for procurement and development functions of the subsidiaries that were in a role of identifying the same components to buy and to use in new products. Anyhow, these meetings had required a lot of resources. Inviting all important stakeholders from all important subsidiaries around the same table from across the country was not a simple task. The meetings were also seen as very slow paced: standardizing even a single component had required a lot of work.

Nevertheless, responsibilities of the data quality were usually seen to belong to the one who created it on a subsidiary level. Participant A3 stated “each function which creates the data should be responsible for the data regarding the parts and components that they use”. Participant A1 added that the company and subsidiaries are generally so small, that there is no possibility that there would be a separate person who would be totally responsible for the master data. On the other hand, subsidiaries noted that this situation-based approach to responsibility had led to some avoidance of the master data quality issues.

The different number of employees also caused a distress to assign unified roles regarding MDM. Subsidiary 1 had lost an important employee who had been responsible for updating the enterprise codes for the purchased components. When the employee left, these responsibilities were transferred to another employee who did not actually have time to do them. Thus, the maintenance of master data was issue and problem based, not preventive. In subsidiary 2, participant C3 stated that the know-how to maintain the data is actually quite well dispersed in the organization so that practically everyone is capable of maintaining the data.

Undefined responsibilities had resulted in many difficulties in the subsidiaries. Sometimes the same job had been done by two different employees at the same time. There had been also cases, when the same component was created twice in the system. Some division of work was seen very necessary in both subsidiaries.

The enterprise initiative to make common master data classification possible with specifically with enterprise classification codes had failed in 2013. One reason the enterprise codes had not been implemented was due to lack of roles and employees. In addition, the subsidiaries had not seen clear benefits for them in the implementation or use of the enterprise codes. The instructions to implement them also lacked some important details, such as how a specific component belongs to a certain classification. The kind of enterprise classification for product components that was initiated, did not contain descriptions about the basis of classifications. Nor did the initiative have ongoing monitoring to follow the implementation.

”A subsidiary is willing to develop master data management practices as long as the other subsidiaries adapt to their practices” Participant A3

There was also stress that the kind of enterprise initiative approach to MDM might not work in future either, nor in long term. Enterprise view, in general, was that the subsidiaries have their reasons to manage their data in a certain way, mainly to ensure the functioning daily operations. These ways were seen as something that the enterprise cannot interfere with because they do not know the specific realities of each subsidiary. Participant A3 concluded this problem in the above-mentioned statement. It seemed as if the creation of a master data initiative would need more communication and co-operation between all the parties involved in the enterprise and all subsidiaries.

5.6 Architectural approaches

5.6.1 Registry style architecture

The registry style was seen as a logical first step towards integration by most of the interviewees, whether they were in an enterprise or subsidiary role. Nevertheless, many practical issues were also raised. The main concern was that a registry table would possibly represent just another isolated data table which needs additional governance, processes, roles, and responsibilities. Important questions raised considering the registry style were: who would create the registry code for each object, how would it be maintained and how can one find the item if it already exists in a database.

On the enterprise level, the registry style raised questions about the benefits and motivations for the subsidiaries to input an additional enterprise code. This question was raised by participants A1 and A3, and it was based on the previous experience to try to implement the enterprise codes. The benefit of enterprise codes was not previously clear for the subsidiaries and it had been either forgotten or de-prioritized due to other work in both subsidiaries. In subsidiary 1, participant B2 stated that the linking of disparate component data, provided by the registry style, would benefit each subsidiary by making inventory management and procurement more managed.

In both of the subsidiaries, there were worries that the registry codes would replace local subsidiary codes. Both subsidiaries had a certain style that they used to create the current codes, and thus a new enterprise code would change a lot practices. When clarifying, that these codes would be an additional linking code, there was again misbelief in the motivations to put the additional code, if there was no clear benefit.

Participant A1 also stated the fact, that the registry style code cannot be the suppliers code, because the supplier changes the component codes on a different basis. The basis to change the codes in subsidiaries was seen mostly when the part was not practically same. While the supplier might change the code of a component when the component changes slightly color, the subsidiaries should be able to use the same code if the component is practically seen the same.

The registry would possibly be maintained in a centralized manner from enterprise or subsidiary, in a leading system manner from one subsidiary or in a decentralized manner from every subsidiary. Participant C5 also noted that ”If every subsidiary has access to add a code, it will end up in even more duplicates”. The registry approach would call for clear roles and responsibilities about what components are updated, by who, how, and how they are inherited to each subsidiary.

At the enterprise level, there were contrasting views whether a leading system approach to registry style could work. Participant A1 stated that the registry style should be started in a way that the one subsidiary system’s data becomes the master data and the other subsidiaries start to use this leading system’s style. However, other interviewees noted that the subsidiaries still have different components, and a leading subsidiary system cannot address most components from the other subsidiaries. So, the problem remained: what would happen to remaining components that are not in the leading system? Other subsidiaries would need access to create master data for their needs as well.

In case the registry style would be implemented as a centralized repository, participant A1 noted that it would still not be possible to identify the similar components and consolidate the master data based on the registry data. Therefore, MDM efforts should start from the largest and most expensive components, which result in most profit. The efforts to consolidate component data would still require physical recognition of components and evaluation whether the component is in the enterprise and subsidiary perspective the same or not.

In the hypothetical case that there would be only one common field also raised concerns about the actual benefits in all organizations. The components would have only one uniform field to see what the other subsidiary has, would still need some knowledge about how the other subsidiary marks their different data fields and it would still allow separate systems and processes. In conclusion, implementing the registry style could cause additional demands, which might overcome the possible benefits.

5.6.2 Hybrid Hub architecture

In a sense, hybrid hub architecture was seen as an extended version the registry style approach, that would not necessarily solve the underlying problem of isolated ERP systems. At the enterprise level, participant A1 thought that the architecture style would still need an expert that would resolve and consolidate the essentially same components. This could be helped, in case the classification metadata would be made uniform because it would make find the components easier. Participant A3 stated that any new effort to make common classifications would not succeed, like previously had happened. At the enterprise level, all interviewees thought making instructions about how to make classifications for different components as a top-down enterprise initiative would probably cause too much resistance in the subsidiaries.

The one way to implement the hybrid architecture approach could be would be a strict top-down rule that would simply force the subsidiaries responsible to make the architecture work. Although this style of management was not presented as something to prefer, it was stated by participant A3 that in case the enterprise would enforce something such as the approach, it should not be implemented in a hurry.

Another learning from the previous trial of implementing the common classifiers was that there should be some ways to actually measure the success. “It has to be so, that the responsibility cannot be avoided”. Another way would be to enforce the input of common identifiers and classifications by having them as a must-have for any new data element that is created in the system.

Regarding the practical solution of data sharing in future, participant A1 stated that it seems inevitable that one subsidiary would become a leading system. The other subsidiaries would follow the leading systems data creation methods, and the new data would be shared from the leading system to the other subsidiaries systems. However, this approach did not take into account that unique component and their new data objects can and should be able to be created from the other subsidiaries as well.

In the hybrid architecture, the creation of new data could happen through a centralized portal common for every subsidiary. When creating a new component in the system, it would occur in a centralized manner, a new enterprise code would be automatically created, and classification would be required. There would thus be a centralized solution to create data elements into each and every one of the subsidiaries isolated ERP systems. Thus, no leading system or a different set of rules would be required for the existing subsidiaries.

It was made clear that there would be many similar problems to the hybrid approach, as in the registry style, such as, how to get the identifying data from suppliers, who would maintain the data, who would decide how the minimum information is put to the system and how to consolidate the instructions to input this data. The degree of standardization would be even higher compared to the registry style, where only common data element was the enterprise identifier, and now it would have some minimum information attached to it as well.

5.6.3 Transactional architecture

The transactional architecture represented the most integrated of the approaches to each subsidiary, and thus, the common reaction from interviewees was that it would require the most amount of work. On the other hand, the general view was that it would result in the most benefits. The enterprise should move towards transactional architecture step by step: first making enterprise codes for the components that would be consolidated, then creating common classifications and specifying minimum information for each, and then consolidating all of the master data under single data management system.

Transactional architecture would also require clear and unified instructions of data creation, technical monitoring of data quality, and some roles and responsibilities to oversee data quality. It was recognized at the enterprise level, that there is a risk that all subsidiaries would use the new system in a similar manner as the old systems, and the challenges would not be solved. Nevertheless, there were concerns about how all of the data from all subsidiaries, which represents thousands of components, would be managed and eventually consolidated into a master data.

Consolidating the data of all subsidiary systems could work in two-phased manner. First, there could be a software solution, or some script to check similar components. When a match with the prescribed amount of similarity would be found, it would be checked by a person if it is truly the same. After being sure, a common enterprise could be added. The ideal end situation would be a common component data with a specific warehouse inventory field for each subsidiary.

The most benefits from this approach would result from putting all of the information in one information system and thus making it easier to search and consolidate all of the data. Participant A3 stated this would be the easiest and would allow consolidation efforts of duplicates into a single data object easier, than the current isolated systems. Nevertheless, the huge work of creating common classifiers and actually consolidating the components would still remain a challenge.

Minimum instructions were seen essential for each category. Some attributes would be optional, some mandatory, but there should be clear rules to overcome this chaotic situation. After common instructions, it would be easier to make the marking processes common. In case these instructions would not be made, the de duplications could be only resolved in procurement and development meetings. In the meeting, they could take initiative to add all important attributes such as enterprise codes, common structures, warehouse and pricing and so forth, to all new components. This approach would require a huge number of meetings and work. These meetings took place currently outside the subsidiary manufacturing plants, so it would not be possible to recognize all components.

In conclusion, none of the architectures in itself presented a perfect solution for the interviewees. It was still acknowledged, that it is good to know what the advantages and problems are in each solution. The enterprise can overcome the problems by addressing them and start developing MDM with some system and processes that remove the most pressing MDM barriers. As long as there exists separate systems storing the same data, there will be many humane errors in typing. Thus, a registry style approach would not work, in the long run, but could work as a starting point. In the case of a new consolidated system, such as a one common ERP, the enterprise might be able to impose some limitations on the attribute inputs technically, have mandatory fields and automated data quality monitoring. A new system that consolidates all of the data was also seen as a point of discontinuity for current data activities. Thus, new instructions could be introduced at the same time as the new system. Consequently, the old ways of working could be abandoned at the same time.

6 DEVELOPING MASTER DATA MANAGEMENT

6.1 Different approaches to developing master data management

MDM addresses the need to create and sustain an organization-wide single version of the truth, a unified data reference, which is utilized in different applications and across business units (Loshin, pp. 9–10). This research was done in a case organization which had an enterprise and multiple subsidiaries. The underlying problem was that the subsidiaries were isolated from each other, used same or similar components, but each had their own way of creating the data, which lead to high degree of redundant data.

Master data initiatives in the case organization were lacking a set of unified goals and practices. These goals are usually considered at the enterprise level but need also to take into consideration individual practices of each subsidiary. The needs for master data management were found to differ substantially between subsidiaries' different functions, between the subsidiaries and between the subsidiaries and the enterprise. The lack of common goals and processes that would support achieving those goals had resulted in multitude of different practices in the creation, distribution, and utilization of master data. The enterprise had presented their own goals to the subsidiaries, but the subsidiaries had not seen the benefits for in terms of their daily operations. There was a gap between global enterprise needs and local subsidiary needs.

The formal goals of MDM should be, for example: aligning with business drivers, formalizing goals through SLAs and quantifying goals in terms of data quality index and data lifecycle time. Functional goals can include data quality, lifecycle, architecture, software tools and training, among data models, processes and application development (Otto 2011). The current situation with MDM goals in the organization was divided. At the enterprise-level, the goals can be described mostly formal. Driving standardization and common procurement among subsidiaries can be seen as aligning business drivers. However, the subsidiaries had functional goals such as data quality, software tools and training. Both levels exhibited goals that related to data models, processes and application development. This "common ground" can possible serve as a starting point for forming common goals for MDM.

Master data management should be seen as a tool that can be used to solve business problems, instead of seeing MDM as an end goal. Initially, the problem of disparate master data was seen mostly as an IT problem and had been approached from data-driven perspective. The underlying business problem seemed to be, how can the subsidiaries use more of the same components in a standardized manner. This, however, could be accomplished through other means and should be considered with a holistic toolset that stems from the acknowledgement that MDM is a business problem, and should reconsider strategies and alignment of business processes as well. Even though the master data would be used across organizations, it might not mean that the components that the data presents will be used in standardized manner.

Considering the organization's limited resources and past failure to address the MDM benefits clearly, it is recommended that the case organization approaches the MDM issue with a low effort solution that would focus specifically on addressing the business impact of master data quality issues. To implement with aforementioned priorities implies a process-driven, problem-oriented strategy (Cleven & Wortmann 2010). Nevertheless, it is important to note, that this approach has also disadvantages: it may not uncover all major master data root problems. Thus, in the long run, when MDM maturity has evolved from its current initial level, it is suggested, that MDM is addressed in data-driven approach as well.

Unified master data standards, classifications and processes should involve both the enterprise and subsidiary perspectives. Previous attempts at making unified classifications were not communicated clearly from the subsidiaries perspective, especially, how this fits into their current processes, and what are the potential benefits, and how each component belongs or does not belong to a certain category. Furthermore, the scope of the classifications might have been too large to realize any quick wins that would demonstrate MDM benefits. The common master data classifications were added to only few products, and it was only utilized briefly by the enterprise in some analytical uses. The subsidiaries had access to such analytical use but had not found any the benefit in using them. However, it is not enough that data rules and handbooks in place, but the information aggregated through master data efforts needs to be available to the local processes (Knolmayer & Röthlin 2006).

Some obvious MDM developments such as global keys were seen in subsidiaries as tedious tasks and ultimately not worth the effort. To overcome the substantial amount of work, it was acknowledged, that there are no magic solutions. The possible benefits of an MDM solution were obvious to most interviewees, but due to limited resources and know-how, it was seen as too much effort. The consolidation efforts were seen as impossible to overcome, because of all of the historical accumulation of disparate data. The creation of new components would be possible to be created in a new, uniform way, but would only represent a fraction of the total data.

The master data maturity of the organization can be described as initial. It was recognized, that some of the data can be called as master data, but it was not utilized effectively in several applications. The analytical usage of master data is characteristic to the initial maturity as well. These were characteristics that can be associated with reactionary level of maturity. Loshin (2010) states that the MDM problem is seen as set of "IT problems" instead of core business problems in the second, reactionary level of maturity. Vilminko-Heikkinen & Pekkola (2017, p. 461) elaborate, that it is difficult to engage the organization in the MDM project, especially business process owners, because they do not understand their role in an "IT project". Reflecting these with the facts that the subsidiaries line of business IT knowledge was initial, it seemed no wonder that previous attempts at master data classifications had failed. Similar classifications can be evolved, when the organization has adapted the fact that MDM represents much more than IT problems.

The case organization was making a transition towards the managed state, where the siloes of the organization are overcome by some level of master data consolidation. MDM had not become a repeatable process. This implies, there are problems on the lower levels of maturity that need to be resolved before proceeding. The enterprise goal of common data structures and classifications for master data seemed very far away, which seems very understandable through this maturity analysis.

When starting to develop MDM, it is important to see it as an evolutionary and continuous process. Combining this with the notion that MDM is a business problem and not an IT problem, it can be safely concluded, that merely developing the architectural layer, or implementing a new system such as a modern ERP, would represent a risky approach. It would represent a bottom-up process. It is not implied, that such approach could not work, as there were major barriers on the levels of systems. Nevertheless, it the MDM development cannot be solved once and for all by just implementing a new solution. In case the organization chooses to develop master data management from bottom-up, that is to say from data and architecture upward, it needs to still consider, how processes, roles and needs are addressed. It is usual, that the adoption of an ERP system generally requires that organization adopts standardized business processes, that are reflected in the design of the software (Hu & Morton 2008).

6.2 Data governance in the organization

Data Governance contains the decision rights and accountabilities for data processes, executed according to agreed-upon models which describe who can take what actions with what information, when, under what circumstances and using what methods (Data Governance Institute 2016). In the case organization, effective data governance would also imply defining the aforementioned in terms of different data domains and different levels of the organization. The aim of data governance is to prevent faulty data in the first place by creating a robust framework for managing interactions between

The geographical distance between the subsidiaries and enterprise had greatly influenced the effectiveness and frequency of governance processes. Initially, master data governance was experienced as difficult to organize. The current practice was that the employees in procurement and development had to be physically present, and it needed a lot of time and resources to organize all important people around the same table. It also took a lot of time for the board to solve problems. Instead of physical meetings, virtual teaming a co-operation with technical means could be utilized. It is suggested, that there should be designated meetings, whether physically or virtually, just for master data issues, and they should be organized more on more frequent basis.

Roles and responsibilities concerning the data quality were not formally defined in any part of the enterprise. The basic roles related to data governance were: data governance director, data governance oversight board, data coordination council and most importantly, data stewards.

The business intelligence manager in the organization somewhat represented the governance director who had capability to monitor all data quality of the subsidiaries. The bi-monthly sales and procurement meetings had a role similar to the data governance board, although, the meeting had multiple other purposes as well and were not seen as effective as they could be, if concentrated only on data issues.

The data stewardship roles, as defined in literature, were completely nonexistent. These roles had been existing in some subsidiaries but had been laid off due to financial problems. This gap represents possibly the biggest opportunity in organizing data governance in the case organization. While it is understandable, that the responsibility of data quality rests mostly on the person who created it. However, it does not make sense how the data quality is measured, when there are no standards or guidelines. Data stewards are needed in between the data owners and users, to note reflect the data against specified standards and situational fitness for use. Only measurement stick currently for the data quality was the flow of internal processes, thus, data was maintained only reactively when problematic situations arise.

It is probable that some subsidiary employees could take the role of business data stewards, who check the quality of data from the perspective of fitness for operational and business use. However, line IT knowledge could be also increased through training to have some technical data stewards as well, who would check the data for technical requirements. It is probable that these roles could not be fully placed in both subsidiaries. Subsidiary 1 might be too small in the number of employees. These roles should be therefore designed case by case. If the subsidiary does not have its own technical data steward, this role can be filled at the enterprise IT level. However, it is important to remember a basic design principle of subsidizing the MDM processes to smallest, lowest and least centralized competent authority (Otto & Ofner 2011). The key enabling factor.is providing stewards the adequate business and IT support, resources and training.

The data stewardship role should include the tasks of trying to maintain the data quality proactively up to date. One approach would be to divide data stewardship roles in different domains of the data. Subsidiary 2 showed some signs of this happening already.

The roles could have not work without data standards and adequate data quality. The data creation happened in only in the subsidiaries, and mostly by the standards that each person had defined on the basis of fitness for operational use. There had been some documented standards for creating the data, but this was rare. A need for documenting was however stated in both subsidiaries. There was a worry at the enterprise level, that how these data standards could be defined. It is suggested that they will not be done at the enterprise level, but at the subsidiary levels in cooperation with the enterprise, and then consolidated into unified and common standards. However, committing to master data criteria and demonstrating commitment is problematic, and should be overcome by demonstrating how the data affects mutual processes (Vilminko-Heikkinen et al. 2016).

Aligning data governance and business strategies is also important. The subsidiaries, as well as the enterprise, were small in size and the subsidiaries had highly related diversification of products, which both might imply centralizing data quality management. On the other hand, a subsidiary's competitive strategy could be seen as a prospector, when considering the degree of innovation of the products. Prospector strategy as well as the locally harmonized processes imply hand a decentralized data quality management. However, it is important to note, that related diversification might enable alignment around a common IT platform (Reynolds & Yetton 2015). In this case, an even tighter functional alignment between different subsidiaries is recommended.

Subsidiaries manufactured similar products under different brand names and had sometimes experiences a competitive situation. This was not currently recognized as a barrier for MDM in the organization. If considering the whole component master data lifecycle, from development to procurement, production and after-sales, the obvious point to overcome master data issues it at the beginning of data creation, the development and procurement. The development has the biggest role in this because they tested and chose the components that were used afterward.

The most pressing factor that would affect the governance design was the tension between two forces that drive change: ingenious development of new products and the component standardization efforts of the enterprise. This is a common problem of globalization versus localization. Considering component master data, this tension will most probably be also reflected in the question which attributes are to be local, and which are to be global. Local attributes should be subject to change flexibly, to cover the needs that subsidiaries face. The global attributes, however, should be created by unified standards, monitored constantly for quality and allow the reliable consolidation of master data elements.

To overcome the strategic tension between separate subsidiaries, one choice would be to manage business processes instead of managing solely data. This could mean a more centralized research and development activities or portfolio management function at the enterprise level. Of course, this was not seen as a valid choice during the research: it was mostly seen that every subsidiary had been and will be autonomous. This does not imply, however, that common strategies in managing the enterprise product portfolio could not be facilitated by the enterprise. These suggestions address already questions of how to align the functions, especially research and development more tightly, and as such, begin to reach out of the scope of this research. This should be also researched by the case organization.

6.3 Master data management architecture

The enterprise architecture in the case organization was an analytical MDM architecture. This solution combined the subsidiary ERP's as source systems for data warehouse and a business intelligence tool as the destination system. The solution was mainly utilized by participant A3. Some development and procurement employees had used the solution, but not found it useful for running daily operations. The data quality was a major barrier to utilizing the analytical solution. The analytical solution had made the data quality problems visible. The analytical tool was accessible to subsidiaries, but not much utilized. Other important systems that stored the master data were the PDM and the ERP systems. These were not integrated in any way, so there was practically no subsidiary-level architecture to analyze.

The lack of subsidiary architecture was a clear barrier to the utilization of MDM. The subsidiaries had basically no architecture, in terms of having integrated applications that use similar data. Thus, the development had used PDM or some simple Excel tool to plan new products, the procurement had created the components by hand to the ERP, and all rest of the functions such as production and spare parts depended on this data.

Inside the subsidiaries, there were no additional applications that were used to store and utilize the component master data. However, the component data residing in the ERP had its predecessors already in the development phase, when it was named in the development drawings or cost of goods calculations. One clear barrier for master data was, that the product development management (PDM) tool had not been fully utilized in both subsidiaries. Also, the PDM was not integrated with the ERP in any way. Also, the component testing library was one important predecessor to the ERP data, data sharing between subsidiaries development functions and ultimately, component standardization. However, the testing had not been structured in an understandable way, and each subsidiary's data was created and structured in folders in a non-standard way.

The current situation with documenting the data standards and classifications a was undocumented. The component data, which was in the focus of this research, was in need of thorough inventory and revisiting the classification rules. Documenting these findings is recommended. The current situation was, that there were only few certain people who knew exactly how and why the components were named and classified the way they were.

The ERP system had some obvious technical limitations, that made it difficult to maintain the master data. Once data was created, it was not possible to remove the components from the database or change some important fields afterward. This had resulted in expired data, which in turn made the inventory lists long and finding needed relevant components difficult. Also, another subsidiary had already filled the limited namespace of some product groups, which made the creation of new component data a complicated process.

There was also too much data in the focus of the scope. The amount of data had made the consolidation and global code efforts seem like a wasted effort. The scope of the MDM initiatives had been too big, they had been focused on the amount of data and the categorizations instead of supporting with actual processes that drive for more consistent data quality.

Local data needs had also not taken into consideration. This was a clear barrier for coming up with practices that fit in the local subsidiary operational processes, but also drive the global master data benefits for enterprise and other subsidiaries involved.

It was recognized, that the organization especially needed a unified enterprise architecture, instead of a disparate subsidiary architecture. Thus, it is recommended that the enterprise would pursue a centralized IT platform in the long run. Despite this clear goal, the consolidation efforts were seen as impossible to overcome, because of all of the historical accumulation of disparate component data. However, only few people mentioned, that the creation of new components would be possible to be created in a new, uniform way.

The conclusion of the architectural research was that the easiest to implement solutions bring also the least of benefits. Thus, the registry style was seen as a good starting point, but with the least benefits. The most robust and also complex system of centralized IT architecture would bring the most results. One of the significant reasons not to implement it was that there would still be a major task of maintaining a global table of master data keys, as it would represent just a registry made for enterprise needs.

The second major conflict considering the architecture was the contrast between subsidiaries local needs and the enterprises global needs. The architectural choice was seen as a major component in balancing these. The architecture of each subsidiary was currently isolated and the resources to maintain an integrated one could be too scarce. Thus, the idea of a single enterprise architecture was seen as a probable solution in the long run. It is suggested in this research also because of the small size of organization and low business line IT-knowledge. Still, the question remained how to transition towards this architecture.

The possible steps toward single enterprise architecture were seen to have influences from all of the architectural styles represented in this research. The registry style was seen as something that could serve as a starting point for MDM. The subsidiaries saw it as a balanced solution. The subsidiaries could still keep their local component codes and run the internal operations normally, on top of the extra global codes. The transition to use global codes could be gradual and could start with a period of having both codes in active use.

However, as Talburt & Zhou (2015) suggest that letting source systems input their own identifier might not work well with component data, which does not have a global code in the source data. Because the local codes are defined on a different basis than the supplier codes, there is no clear global identifier in the source data. Thus, it would make sense, in case the global code would be defined by the enterprise practices, and not let to the local source systems. Other way, which had been already attempted with some suppliers, was to attempt to guarantee that the suppliers will provide the same global code for subsidiaries and then let the subsidiaries input the global code.

The next step, implemented at the same time or after having global component identifiers could be to implement some global classifications. This would further improve the processes of consolidating the old master data. After common classifications, it would be easier to define a common enterprise data model and transform the component data into one common transactional architecture, such as a common ERP system, while still keeping local codes and attributions in place to ensure a frictionless transition.

Even though the enterprise would implement a transactional architecture, it might be too strict for the subsidiaries to have all attributes unified. This is especially the case when it is important to let the subsidiaries have their current operational processes intact. Firstly, it is suggested that the underlying business needs of those processes would be evaluated. Then, the transactional system should be approached flexibly, meaning that some attributes could be left to as subsidiary-specific local attributes, that are needed to enable the subsidiary-specific business processes.

However, the high degree of relatedness between the subsidiaries products applies also to domain of processes. The development, procurement, production and after sales functions seemed to work with very similar goals and motivations. This implies, that it could be useful for subsidiaries to also share the know-how they have on different processes. The component data and the processes that utilize it might be both developed concurrently to achieve the right balance.

The previous points have a lot to do with how new data is created during its lifecycle. However, the MDM problem is even more about how to consolidate the data that already exists. The previously mentioned points such as data standards, classifications and target architectures address this problem. However, the entity resolution contains the actual parsing, standardization, transformation and matching which is made to the accumulated data.

The entity resolution represented an impossible obstacle for some participants, but it can be approached in many ways that will provide benefits. Some of the possible hundreds of duplicate components need to be consolidated by hand, but it is suggested, that the enterprise would evaluate the possibilities to half-automatically match duplicate data. It is assumed, that the matching could be made easier by first standardizing some classifications across the subsidiaries and then using matching algorithms or at least narrowing down possibilities.

Entity matching cannot be done at the enterprise-level, because they lack the specific component knowledge. On the other hand, the matching cannot be done at the subsidiary-level, because the subsidiaries lack IT-knowledge. It is suggested that this gap knowledge would be solved either by adequate training to use an entity matching tool or solution, and with a cooperatively designed process that is clearly linked to the daily operations and goals of a subsidiary. The process should also consider false positives and negatives, and the possibility to monitor matched duplicates and have a “roll-back” if needed.

To alleviate employees feeling overburdened with the consolidation, it should be analyzed which are the most useful components to be matched and prioritize them. Choosing a specific domain of specific bought components i.e. components that are used in variety of products, which are the most expensive, or which are used in general across many subsidiaries, might be some criteria for prioritization. Also, keeping in mind that the cost of master data should be overcome by the benefits of common master data. The cost of master data should be thought of as a long-term investment, which might bring increased returns through efficiency and standardization, but also by eliminate the hidden costs of bad data quality.

7 CONCLUSIONS

7.1 Summary and key findings of the thesis

Data is considered as one of the important assets of an organization and needs to be managed accordingly. Master data management (MDM) attempts to create and sustain an organization-wide unified data reference, which is utilized across business units. In the case organization, the attempt was to create a unified component data reference that would be utilized across subsidiaries and their applications. The underlying problem in the master data development was that the subsidiaries used somewhat the same or similar components, but each created their unique data and classifications, resulting in a large number of potential duplicates.

MDM should not be IT problem, but a business problem that spans people, processes, and technologies. It requires the whole organization define common needs, goals, data standards, systems, roles, processes and monitoring. MDM cannot be solved over one project, but it is a continuous process of responding to changing business needs. The ideal state of MDM is when master data can be utilized strategically. This was also the case organizations aspiration, although the more immediate goals involved efficient reporting and data sharing in general.

Master data management necessitates also governance to effectively initiate and maintain the decision rights and accountabilities for each role, process and data domain. Data stewards, who are situated functionally in between IT and business, source and target systems in terms of architectural style, or data owners and users in terms of the people in organization, are the essence of master data management. They cover the whole range of MDM to ensure the fitness for organizational needs. The goals and needs are monitored and developed at the levels of teams or coordination councils, depending on the organization.

Although MDM is not an IT problem, a big part of the MDM challenge are disparate information systems. The case organization had also faced this challenge, and on top of that noticed that the current systems are outdated in terms of efficient update processes. The enterprise architecture was downstream analytical, utilizing the subsidiaries systems as sources. However, the architectural style must also be aligned closely with the formal and functional needs.

There were many barriers to developing master data management, which also touch all sides of organizational development. The current state of master data management was studied in the empirical part of this research using case study methods. Stakeholders attended interviews from the organization and two subsidiaries. The barriers that were elaborated in this case study included: not recognizing the importance of master data for long term organizational success, misalignment between business and IT strategies, tension between formal and functional requirements, undefined roles and responsibilities, undefined data standards and classifications, lack of monitoring, lack of flexibility and features in the master data system.

The suggestions for development were grouped under three themes: different approaches to develop master data management, data governance and master data management architecture and systems. The themes and key points, as well as suggested actions are concluded in table 7.1 below.

Theme	Key point	Suggested actions
Different approaches to develop master data management	Lack of common goals between enterprise and subsidiaries	→ Considering both formal and functional needs when designing MDM processes, roles and architecture. → Defining goals for enterprise, subsidiary and for whole organization.
	MDM should be seen as a tool	→ Develop business processes and best practices in alignment with architectural development.
	Misaligned business and IT strategies	→ Develop alternative ways to drive the underlying need for component standardization, such as best practice sharing and functional alignment between subsidiaries
	MDM is seen as too big, "one-off" project	→ Start small with a specific domain of data and present the value of MDM fast, develop continuously → Track and evolve slowly in stages of maturity
Data governance in the organization	Lack of defined roles and responsibilities	→ Define the accountabilities and decision rights, assign roles and decide their position in the organization → Assign and train data stewards for important domains
	Lack of unified standards and processes	→ Define data standards in cooperation, communicate them effectively and demonstrate value to subsidiaries → Give decision rights at the lowest possible level of organization
	Lack of monitoring	→ Define metrics and monitor them regularly → Have automatized checks where possible
	Practical barriers to organizing	→Utilize virtual team meetings and teams for organizing MDM when face-to-face meetings are not possible
Master data management architecture	Loosely integrated subsidiary systems, no enterprise architecture	→ Define which systems are to be part of enterprise architecture and which are to the subsidiaries → Integrate systems that contain component data (PDM, ERP) → Prepare for consolidation with common classifications
	Discrepancies between subsidiaries	→ Define which attributes are to be global and which local → Seek for alternative ways to make competitive strategies aligned, for example portfolio planning
	Out-dated ERP system	→ Evaluate possibilities for new system and possibly organizing around common IT platform → Redesign common business processes if platforming

Table 7.1. *Summary of key findings and suggested actions to develop master data management.*

7.2 Managerial implications and suggestions

The empirical part of research is based on interviews that were gathered in December 2016, and therefore, it is important to note that many things have happened in the organization after that. Some steps have probably been taken towards evolving in MDM maturity, and thus the suggestions presented in this research need to be evaluated in the light of current situation. Nevertheless, these recommendations, should be noted. The problem of master data management is complex, especially in multi-business environment, where there are additional barriers to effective cooperation.

It is important to note, that much of the academic literature around master data management, and especially case studies are focused on large enterprises. For example, the case organization might not have enough employees to place each role that is needed in the data governance framework to a different employee. The literature might have assumptions, that organization has a separate IT department in which it would be convenient to place some updated data quality activities. It is unlikely so. However, these perspectives should be kept in mind, while reading this thesis. It was not possible to limit the research scope to small and medium-sized companies, due to the fact that there is not much literature to be found.

As stated in this research MDM is just a tool, but not the end purpose. It was clear during this research, that master data definitions, roles, processes, and architecture are just means to an end. The underlying end purpose and more implicit business problem seemed to be about how to increase cooperation and information sharing that is likely in the common interest of all stakeholders in a the very specialized domain of products that the organization was engaged in. Instead of seeing this underlying problem as an IT problem, and not a business problem, might have limited the ”tools of choice” and the underlying problem statement too much into the area of master data and architectures. Thus, placing value more on organization-wide functional cooperation and common goals instead focusing too much on common systems and sophisticated architectures is suggested.

7.3 Limitations of this research and suggestions for future research

MDM development has not been extensively studied in the context of multi-business organizations, although there is a lot of literature about MDM itself, data governance, and IT-business alignment in multi-business organizations. This thesis is considered as somewhat revelatory case in the way it combines and applies the theoretical background from these various domains. Also, many MDM situations in real life present a similar problem that is described in this research: master data problems arise when multiple businesses (or lines of business) have disparate systems.

Early on in the research process, it was noted that it was difficult to find the right balance between the organization's needs for research and the academic requirements for writing a thesis. However, this balance was found in the end through the case study method, which allowed the research to include broad range of subjects and go deep into the organizations daily processes. As a case study, this thesis has obvious limitations in terms of generalizability. From the perspective of the organization, this embedded case study is not unfortunately complete, as it had to leave out one of the three subsidiaries.

In terms of what qualifies as an exemplary case study, this thesis is seen as adequate. The research questions were seen to fit very well to the scope of the case study. It investigated a phenomenon in depth within the real-life context, and the boundaries between phenomena and context were not clearly evident. Understanding the subsidiaries' needs would have needed even deeper understanding of the highly contextual conditions, under which data is created, maintained, utilized and shared. In this perspective, another round of free-flowing interviews could have been helpful to have an exhaustive collection of data. However, the thesis is seen as satisfactory, because of the breadth of themes and rivaling propositions in the empirical part compensate this drawback.

Exemplary case study is technically sound when it maintains the chain of evidence and produces insights into human processes (Yin 2008). This point is left evaluated to the reader. It is seen, that the thematic analysis used to process the empirical data lead to actionable insights. The case study is seen of some general public interest, because master data problem is quite common situation in organizations with multiple business units, siloed functions or, like the case organization, many acquired subsidiaries. MDM is not a popular topic of discussion although, there is growing media interest for developing machine learning and artificial intelligence solutions, which are without exception based on well managed data.

Further research could inquire, how the small size of organizations relates to master data development. Such academic literature seemed to be scarce. Furthermore, the organization might want to do further research with narrower scope about how the data governance can be effectively aligned in organizations, especially when business units have high relatedness diversification and who are still geographically apart. This could be beneficial to study also for similar organizations who have grown through acquisitions and have similar MDM problems.

BIBLIOGRAPHY

Ackoff, R., 1989. From data to wisdom. Journal of applied systems analysis, Vol. 16, Iss 1, pp. 3–9.

Allen, M. & Cervo, D. 2015. Multi-domain master data management: Advanced MDM and Data Governance in Practice. Morgan Kaufmann. USA.

Bollinger, A.S. and Smith, R.D., 2001. Managing organizational knowledge as a strategic asset. Journal of knowledge management, Vol. 5, Iss. 1, pp. 8–18.

Bhansali, N. 2013. Data Governance: Creating Value from Information Assets. CRC Press. USA.

Brou, P., Janssen, M. and Vilminko-Heikkinen, R. 2016a. Coordinating decision-making in data management activities: A systematic review of data governance principles. International Conference on Electronic Government and the Information Systems Perspective, pp. 115–125.

Brou, P., Pekkola, S., Vilminko-Heikkinen, R. 2016b. Paradoxed, conflicts and tensions in establishing master data management function. ECIS 2016 Proceedings, Paper 184.

Chen, D., Doumeingts, G. and Vernadat, F., 2008. Architectures for enterprise integration and interoperability: Past, present and future. Computers in industry, Vol. 59 Iss. 7, pp. 647–659.

Christen, P. 2012. Data matching: concepts and techniques for record linkage, entity resolution and duplicate detection. Springer. USA.

Cleven, A. and Wortmann, F. 2010. Uncovering four strategies to approach master data management. System Sciences (HICSS), 2010 43rd Hawaii International Conference on 5.1.2010, pp. 1–10

Dahlberg, T. 2015. The MDM golden record is dead, rest in peace – welcome interpreted interoperable attributes. Åbo Akademi, pp. 1–13.

Duff, A. 2005. Master data management roles-their part in data quality implementation. 10th International Conference on Information Quality.

Ebner, V., Otto, B., Österle, H. 2012. Conceptualizing data in multinational enterprises: model design and application. Proceedings of Conceptual Modeling: 31st International Conference, ER 2012 Florence, Heidelberg, October 15–18, Lecture Notes in Computer Science, Vol. 7532, pp. 531–536.

Darke, P., Graeme, S., Broadbent, M. 1998. Successfully completing case study research: combining rigor, relevance and pragmatism. Information systems journal, Vol. 8, Iss. 4, pp. 273–289.

Dreibelbis, A., Hechler, E., Milman, I., Oberhofer, M., van Run P., Wofson, D. 2008. Enterprise Master Data Management: An SOA Approach to Managing Core Information. IBM Press. USA.

Godinez, M. Hechler, E., Koenig, K., Lockwood, S., Oberhofer, M., Schroeck, M. 2010. The Art of Enterprise Information Architecture: A Systems-Based Approach for Unlocking Business Insight. IBM Press. USA.

Hamel, G., Prahalad, C. 1990. Strategic intent. Harvard Business Review, Vol. 68 Iss. 3, pp. 18–38.

Haug, A., Arlbjørn, J., 2011. Barriers to master data quality. Journal of Enterprise Information Management, Vol. 24, Iss. 3, pp. 288–303.

Hu, Q., Morton, N. 2008. Implications of the fit between organizational structure and ERP: A structural contingency theory perspective. International Journal of Information Management, Vol. 28, Iss. 5, pp. 391–402.

Khairi M., Shaykhian, G., 2014. Factors influencing data management model selection. 121st ASEE Annual conference & exposition, June 15-18 2014, Indianapolis.

Knolmayer, G., Röthlin, M. 2006. Quality of material master data and its effect on the usefulness of distributed ERP systems. ER Workshops 2006, LNCS 4231, pp. 362–371.

Ladley, J. 2012. Data governance: How to design, deploy, and sustain and effective data governance program. Morgan Kaufmann. USA.

Lee, A. 1989. A scientific methodology for MIS case studies. MIS Quarterly, Vol. 13, Iss. 1, pp. 33–50.

Loser, C., Ledger, C., Gizanis, D. 2004. Master data management for collaborative service processes. International Coherence on Service Systems and service management. Beijing 19-21 July.

Loshin, D. 2010. Master Data Management. Morgan Kaufmann. USA.

Maedche, A., 2010. An ERP-centric Master Data Management Approach. AMCIS 2010 Proceedings, Paper 384.

Marsh, R. 2005. Drowning in dirty data? It's time to sink or swim: a four-stage methodology for total data quality management. Database Marketing & Customer Strategy Management, Vol. 12 Iss. 2, pp. 105–112.

McGilvray, D. 2008. Executing data quality projects: ten steps to quality data and trusted information. Morgan Kaufmann. USA.

Mosley, M. 2010. The DAMA Guide to the data management body of knowledge. 1st edition. Technics Publications. USA.

Nedumov, Y., Turdakov, D, Maiorov, V. and Ovchinnikov, P. 2013. Automation of data normalization for implementing master data management systems. Programming and Computer Software, Vol. 39, Iss. 3, pp. 115–123.

Otto, B. 2011. A Morphology of the organization of data governance. ECIS 2011 Proceedings, Paper 272.

Otto, B., Schmidt, A. 2010. Enterprise master data architecture: design decisions and options. Little Rock. USA.

Otto, B., Ofner, M. 2011. Strategic Business Requirements for master data management systems. AMCIS 2011 Proceedings, Paper 108.

Radcliffe, J. 2007. The Seven Building Blocks of MDM: A Framework for Success. Gartner research, October 2007.

Reichert, A., Otto, B., Österle, H. 2013. A Reference Process Model for Master Data Management. 11th International Conference on Wirtshcaftsinformatik. Germany.

Rindfleisch, A., Moorman, C. 2001. The acquisition and utilization of information in new product alliances: Strength of ties perspective. Journal of Marketing Vol 65, Iss. 2, pp. 1–18.

Sambamurthy, V. and Zmud, R. W. 1999. Arrangements for Information Technology Governance: A Theory of Multiple Contingencies. MIS Quaterly, Vol. 23, Iss. 2, pp. 261–290.

Saunders, M., Lewis, P., Thornhill, A. 2016. Research Method for Business Students. 7th edition. Pearson Education Limited. England.

Silvola, O., Jaaskelainen, H., Kropsu-Vehkaperä, Haapasalo, H. 2011. Managing one master data – challenges and preconditions. Industrial Management & Data Systems, Vol. 111, Iss. 1, pp. 146–162.

Talburt, J., Zhou, Y. 2015. Entity information life cycle for big data: Master data management and information integration. Morgan Kaufmann. USA.

Tuomi, I., 1999. Data is more than knowledge: Implications of the reversed knowledge hierarchy for knowledge management and organizational memory. Systems Sciences. Proceedings of the 32nd Annual Hawaii International Conference IEEE.

Vilminko-Heikkinen, R. & Pekkola, S. 2012. Organizational issues in establishing master data mangement function. Proceedings of the 17th International Conference on Information Quality, 16-17 November 2012, Paris, France, pp. 1–13.

Vilminko-Heikkinen, R. & Pekkola, S. 2017. Master data management and its organizational implementation: An ethnographical study within the public sector. Journal of Enterprise Information Management, Vol 30, Iss 3, pp. 454–475.

Waddington, B. 2010. Data governance, MDM and data quality. Information Management Vol. Sept/Oct, Iss. 10, pp. 14–16.

Wand, Y. Wang, R.Y., 1996. Anchoring data quality dimensions in ontological foundations. Communications of the ACM, Vol 39, Iss. 11, pp. 86–95.

White, A., Newman, D., Radcliffe, J., 2006a. How to choose the right architectural style for master data management. Gartner Research. September 2006.

White, A., Newman, D., Logan, D., Radcliffe, J. 2006b. Mastering master data management. Gartner Research. January 2006.

Weber, K., Otto, B., Österle, H. 2009. One size does not fit all – A contingency approach to data governance. ACM Journal of Data and Information Quality 1, No. 1. June 2009.

Wende, K. 2007. A model for data governance - organising accountabilities for data quality management. 18th Australasian Conference on Information Systems, pp. 417–425.

Yin, R. 2008. Case Study Research: Design and Methods. 4th edition. Sage Publications. USA.

APPENDIX A: INTERVIEW QUESTIONNAIRE

Creating data (presented only to ones in position to create data)

1. When is new component data created?
2. When should new component data be created?
3. In which specific situations should a new component data be in the system?
4. What attributes are important to input in the system?
5. Which information sources are used when creating the data?
6. In what format is data input?
7. Which attributes are especially important (for the function of interviewee)?
8. Which component attributes should be same across the subsidiaries?
9. What conditions should there exist to match real life components, based on data?
10. Why is the component data input in the specific way it currently is?

Maintaining data

11. Who is responsible of data quality in the organization?
12. How is data maintained in the organization?
13. Why is data maintained?

Utilizing data

14. What data do you utilize?
15. How do you utilize data?
16. What challenges/barriers you have faced while utilizing data?
17. How could you utilize data better?
18. What changes should be implemented to better utilize data?

Roles and responsibilities related to data

19. Who is responsible of the data quality of the created data?
20. Who is responsible of updating the data?
21. How are the roles and responsibilities defined?
22. What challenges/barriers have you faced with how roles and responsibilities have been defined?
23. How have these challenges been addressed in the past?

Sharing data

24. What data you share across subsidiaries?
25. What data would you want for other subsidiaries to share to yours?
26. How is data shared across subsidiaries?
27. How does sharing data across subsidiaries currently work?
28. Why sharing data across subsidiaries is challenging/rewarding?
29. How could data be shared more efficiently across subsidiaries?

Data architecture

30. How changing the practices in data would affect the organizations operations?
31. How a centralized architecture would affect the organizations operations?
32. How a hybrid hub would affect the organizations operations?
33. How a transactional architecture would affect organizations operations?