DOI: 10.15514/ISPRAS-2025-37(4)-9

Managing Master Data Implementations

S.V. Kuznetsov, ORCID: 0000-0001-6752-6742 <sergey.kouznetsov@gmail.com>
D.V. Koznov, ORCID: 0000-0003-2632-3193 <d.koznov@spbu.ru>
D. V. Luciv, ORCID: 0000-0002-6332-2360 <d.lutsiv@gmail.com>

Saint Petersburg State University
7/9 Universitetskaya emb., St. Petersburg, 199034, Russia.

Abstract. Every business organization has a subset of data which must be highly consistent: legal information, supplier and contractual data, customer base, etc. Customers and employees expect to receive the same information about the same data object from different organization sources, which are usually other information systems. The process of consolidation and centralized control of such data throughout the organization is called Master Data Management (MDM). The iterative deployment strategy is a popular way to introduce MDM to a organization that supposes a step-by-step implementation of MDM components based on the real needs of the organization. In this paper, we present a functional MDM model for the early stages of MDM implementation within the iterative deployment strategy. The purpose of this model is to represent real business needs of an organization in terms of MDM, making clear which MDM components should be implemented, and which should not. Detailed description of the model components is provided. Also, a case study, presenting a portfolio of six real MDM projects analyzed from the viewpoint of the proposed model is performed.

Keywords: enterprise applications; master data management; open source; component approach.

For citation: Kuznetsov S.V., Koznov D.V., Luciv D.V. Managing Master Data Implementations. Trudy ISP RAN/Proc. ISP RAS, vol. 37, issue 4, part 1, 2025. pp. 161-176. DOI: 10.15514/ISPRAS-2025-37(4)-9.

Управление MDM-проектами

С.В. Кузнецов, ORCID: 0000-0001-6752-6742 <sergey.kouznetsov@gmail.com>
Д.В. Кознов, ORCID: 0000-0003-2632-3193 <d.koznov@spbu.ru>
Д.В. Луцив, ORCID: 0000-0002-6332-2360 <d.lutsiv@gmail.com>
Санкт-Петербургский государственный университет,
Россия, 199034, Санкт-Петербург, Университетская наб., д.7-9.

Аннотация. Управление мастер-данными (Master-Data Management, MDM) – особый вид управления данными бизнес-организации, нацеленный на идентификацию, очистку, консолидацию и централизованное управление важнейшими данными компаний (golden dataset), которые обычно распределены по разным информационным системам и другим источникам. Ведущие мировые поставщики программных решений (IBM, Oracle, Informatica и многие другие) предлагают широкий спектр готовых продуктов по управлению мастер-данными (MDM-продукты). Однако внедрение MDM сопряжено с большими затратами: необходимо не только адаптировать эти продукты под особенности организаций, но и выполнить изменение бизнес-процессов, создать новые политики работы с данными, решить вопросы безопасности, решить другие вопросы. В связи с этим в России популярна итеративная стратегия внедрения MDM, подразумевающая поэтапную реализацию управления мастер-данными на основе реальных нужд организации-заказчика. В работе вводится понятие MDM-решения, которое является результатом внедрения MDM в организацию и включает помимо программной части также обработанные мастер-данные, налаженные организационные процедуры по сопровождению мастер-данных и прочее. Основным результатом статьи является функциональная модель управления мастер-данными, предназначенная для ранних стадий разработки MDM-решения в рамках итеративной стратегии. Целью данной модели является отобразить реальные потребности организации на язык MDM – важно понять, какие именно MDM-компоненты должны быть реализованы в первую очередь. В работе также приводится экспериментальное исследование, в котором уже реализованные MDM-решения анализируются с помощью предложенной модели.

Ключевые слова: корпоративные информационные системы; мастер-данные; открытые системы; компонентные системы.

Для цитирования: Кузнецов С.В., Кознов Д.В., Луцив Д.В. Управление MDM-проектами. Труды ИСП РАН, том 37, вып. 4, часть 1, 2025 г., стр. 161–176 (на английском языке). DOI: 10.15514/ISPRAS–2025–37(4)–9

1. Introduction

Converting to digital economy requires business data management in to be as universal as, for example, accounting [1]. It is impossible to imagine a normally functioning organization that does not properly support special data such as legal information, supplier and contractual data, a customer base, etc. Its users rely on its consistency within the organization, i.e., they expect to receive the same information about the same object from different sources, which are mainly the information systems within the organization. Inconsistencies and contradictions in the data lead to problems, delays, and collisions, as well as financial and public image losses [2]. This special data, when consolidated and properly maintained, is commonly called Master Data[1], and, consequently, the process of its consolidation and maintenance is called Master Data Management (MDM) [1, 8].

[1] Reference data, such as registers, classifications, directories, etc., is often distinguished as a special type of master data. In Russia, handling of reference data is highly developed: there is a large volume of such data in large organizations, as well as a wide set of standards (federal, industrial, and regional) and solutions [3–7].

MDM includes a wide set of methods and strategies which have been captured and listed in a structured fashion within the DAMA-DMBOK (Data Management Body of Knowledge) [2]. There is also a large number of ready-made MDM products by well-known vendors such as IBM, Oracle, Informatica, etc. (see, for example, Gartner's Magic Quadrant report 2021 [9]). However, MDM implementation is currently not standardized and well-elaborated, since its target organizations, especially large ones, possess a lot of various individual features.

In practice, there are two fundamentally different strategies of MDM implementation: top-down and iterative [2]. The top-down strategy supposes the following chain of actions: creating a strategic conceptual MDM framework for the organization, specifying the MDM requirements, customization and fine-tuning already existing MDM products, performing necessary administrative work, and finally, running and maintaining the MDM solution. The iterative strategy, in its turn, supposes that MDM is implemented to solve a specific important problem of the organization, assuming the following extension of the MDM functionality and/or implementation of MDM for other businesses of the organization, i.e., to solve other business cases. These strategies correspond to the types of the organizational innovation proposed in [10]. The first one can be compared to technology push: innovation consists of the integrating cutting-edge technology intended to solve various, including even currently unknown, problems within the organization. The second one is comparable to the organization pull: the organization, or more precisely, its specific needs, initiate the innovation. Not rejecting the first strategy, we focus on the second one. It is less risky and allows to achieve specific practical results within manageable timeframe.

During iterative deployment of an MDM project, the organization's requirements need to be translated into the MDM terminology, which is highly advanced and has shown considerable formalization progress [2]. If these needs of the organization translate well into MDM, then it can be solved using existing MDM tools [9], which significantly cuts the costs of the project. However, it is often the case that the organization is not well-versed into MDM terminology, and tries to propose other types of projects or request an implementation of an MDM project from the scratch. Such mistakes lead to collisions, extended deadlines, and financial loss.

In this article, we propose a functional model of master data management that is intended to support the process of preliminary estimation and coordination of the work in the iterative MDM projects at their earliest stages. Note that we are primarily focused on creation/deployment of the IT infrastructure that supports MDM, as well as setting up and performing necessary analytical work (data cleaning and consolidation, classification and hierarchization, etc.). Further on, the organization will perform this analytical work on a regular basis using the deployed infrastructure, and its complete automation should be the ultimate target. Focusing on the software and analytical aspects of MDM, we purposefully omit important aspects, such as the modification of the business processes and data handling practices, as well as employee training and other issues. In our experience, they are solved much better when MDM software is fully deployed and implemented. Furthermore, there is a discipline dedicated to these topics – Data Governance [11]. Below we use the proposed model to describe several industrial MDM projects that we have participated in, demonstrating its applicability in practice.

The approach was briefly outlined in [12]. In this paper, we explain our ideas in more detail, including adding new examples. Moreover, the experimental study was extended with additional attributes of MDM projects, namely, overall data volume in a company, master data volume, number of data sources, project work distribution (software development, analytical work, and data processing).

The rest of the paper is organized as follows. Section 2 presents a brief overview of the master data concept. Section 3 provides the definition of the master data management projects. We introduce the MDM solution in Section 4 as a result of an MDM project, and describe its various compounds. The proposed Functional Master Data Management Model is described in Section 5. In Section 6, we present a case study with completed real-life industrial MDM projects, in which the proposed

model was employed. Section 7 contains an overview of the related work. Finally, we present our conclusions in Section 8.

2. Master Data

Following the Gartner Glossary [8], master data is a special type of business data that describes most essential characteristics of an organization, such as its current and potential customers, suppliers, consumed and manufactured products, office and production sites, billing information and details concerning accounts of individuals or counter-parties that it deals with, etc. The point of distinguishing master data as a separate concept is that medium and large organizations possess variety of information systems and use many external heterogeneous data sources. In the result, the same information ends up being represented by different data and in different formats. Furthermore, various sources may contain different attributes for the same data, and the data itself may be contradictory. In general, this can be tolerated to certain extent, but there is a subset of data which cannot tolerate disorganization at all: critical data, inconsistencies in which harm the organization and inhibit its normal functioning. This particular type of data receives special treatment thus becoming master data.

Therefore, an organization should launch a dedicated process of *Master Data Management (MDM)* for efficient consolidation, usage and maintenance of the master data. MDM supposes that a organization properly cares about its business processes, quality control and integration of data, as well as standardization of the existing information systems [13]. In other terms, it focuses on collecting and accumulating data from various sources, i.e., information systems that exist within the organization (data sources, DS), additional consolidation of this data and its distribution (delivery) to the consumer information systems (data consumers, DC).

DAMA methodology [2] identifies the following key steps of an established MDM process within an organization: (i) data model management, (ii) data collection and accumulation, (iii) data validation, standardization, and enrichment, (iv) entity and data inconsistency resolution.

MDM should be supported by a special IT solution created and implemented within an organization (further referred to as an *MDM solution*). An integral part of an MDM solution is a central repository or *data hub*. It collects master data candidates which are then appropriately processed and delivered to DC systems. Gartner identifies four approaches to the Data Hub architecture [9]:

- *Registry*. The hub does not contain the data itself, but only the corresponding references (indices). This approach is relevant for the data that cannot be copied or "moved" for various reasons, e.g. data under certain regulation.
- *Consolidation*. Data is uploaded into the common repository on a regular basis, appropriately processed, and then the hub itself provides DC systems with an access to this data. Here new data is uploaded into the hub on a regular basis by DS.
- *Centralization*. This architecture is very similar to the previous one, but the hub takes over data input as well: i.e., data could be input directly to the hub itself, and thus turning all systems that initially were DS into DC.
- *Coexistence*. This architecture implements a combination of the Consolidation and Centralization for different master data of an organization, i.e. some information systems could play both roles of DS and DC. Additionally, if some data fragments are not "movable", they can be handled using the Registry approach.

A wide set of ready-made software tools to create MDM solutions already exists. First of all, there are so-called "boxed" products such as SAP MDG, Informatica MDM, IBM InfoSphere MDM and others, which are focused on solving standard tasks of master data management. However, these tasks can vary to a large extent. This has motivated some vendors, including some of the abovementioned ones, to offer "software construction kits" which can be pieced together for a given need, for example, Informatica MDM, Unidata, and others. Thus, implementing an MDM solution for a specific organization remains a complicated and labor-consuming effort.

3. MDM Projects

In practice, MDM projects often start as common IT projects. Therefore, it is necessary to identify whether the given needs of an organization are clearly MDM-oriented, which allows using ready-made MDM products.

An organization clearly needs MDM when it finds necessary to perform data collection, enrichment and consolidation from various DS systems, as well as the delivery of this data to various DC systems, which then use it to support various business functions. DS can be both internal and external: for example, an organization may need to enrich its customer data with information collected from social media. The requirement for several DCs receiving master data is less strict and sometimes may be omitted. However, in this case the organization runs a critical business process that requires high-quality enriched consolidated data obtained from various DS systems. For example, processes such as validation of a new client or of a suspicious transaction in a bank.

Let us also consider when a organization does not need an MDM project. Firstly, when the processed data is homogeneous: for example, it was entered into an information system or set of systems manually by operators (operator input). These projects may require a logical data model, data cleaning and validation, various modes of data access, and so forth. But they lack the main MDM task of the data consolidation from different sources. Secondly, when complicated business logic is required: this type of functionality should be moved out of the MDM solution into separate information systems [9]. An MDM project ends at the delivery of the master data to DCs.

In summary, MDM-specific tasks must be the central focus of the project. If it is not the case then this particular IT project is not an MDM one.

4. MDM Solution

As a result of an MDM project, the organization receives an *MDM solution*. It includes the following: (i) implemented MDM-product as *MDM-system* (the software), (ii) new data handling rules, (iii) trained employees, and (iv) an established and running Master Data Management process. The last point is of the same importance as the other ones: the business process may not start following MDM process due to other problems, e.g. security issues, high workloads of employees that were assigned to with various MDM-specific tasks in the process, etc.

Let us describe in detail what an MDM system is. First of all, it is deployed software system that implements the main MDM functionality: data hub, consolidation and survivorship rules, etc. The main part of this software system is the customized MDM product (see the list of the available products in [9]). Additionally, it may include a set of utilities and various tools that perform particular tasks (such as data cleansing). Availability of the ready-made multifunctional software, which requires just configuration and deployment within the organization significantly decreases the costs and risks of the project. However, some compounds of an MDM system have to be implemented separately within the MDM project in order to represent the specifics of the organization that could not be covered via standard tools. MDM is commonly implemented in large and complex organizations that have already established their own processes and obtained their own unique traits. Therefore, a completely ready-made software solution that is suitable for any given task or business case does not exist and MDM products need to be modified to meet all organization needs.

5. Model Description

It is necessary to perform more detailed analysis is required after a preliminary informal discussion of the organization's needs in order to find out how to precisely express them via MDM [8, 9]. We propose a special functional model to increase the efficiency of this process. It describes a typical MDM solution, including an all-encompassing functionality set, so that its users are able to choose which functions they need to implement to serve their particular needs. The proposed model can be

called an ideal model to-be: this term is common in structured [14] and object-oriented analysis [15], as well as in business process re-engineering [16].

For the ease of use, the model is represented via a full lifecycle metaphor of master data. In software engineering research, a metaphor is an analogy drawn with physical reality in order to elaborate on virtual abstractions. In this case, we will use a model life cycle of master data as a metaphor, keeping in mind that a real life cycle is a much more complicated process. However, the package-steps that we have identified allow to conveniently structure the functional components of a typical MDM solution and use the model for planning the functionality of the specific MDM solutions.

Our model consists of three packages (steps): data collection, data processing, and data delivery. Packages contain functional components each describes a set of activities concerning setup or management of the master data. Therefore, components of our model include both MDM implementation and its further support, i.e. the activities performed during the following functioning of the MDM solution.

For example, in order to implement Data Consolidation, the following activities are required:

- MDM implementation: customize/configure existing software that supports the analyst's workplace, define the rules for conflict resolution, and consolidate the data from organization DS systems for the first time.
- MDM support: further data consolidation as one of the MDM solution functions, since the data from DS systems will continue to flow to the data hub.

Discussing MDM projects, we strictly focus on *software development* and *analysis* aspects of MDM. The software part of an MDM project includes customization of a ready-made MDM product, which can be configured and modified for the organization's specifics, and development of the special software for some particular features such as access to data of existing information systems in an organization, data quality scripts (e.g. special cleaning rules which are applicable for this particular data of this particular organization), software to resolve consolidation conflicts (e.g. introducing appropriate machine learning algorithms), etc. Analysis aspects refer to the establishment and initial performance of the analytical work: data cleansing and consolidation, classification and hierarchization, etc.

It should be noted that we omit other activities such as business processes modifications, establishment of new data policies, employee training, etc. On one hand, the latter are usually performed by the organization itself, as the MDM project is implemented within the iterative strategy, and therefore is an answer to the organization's specific request. Thus, the organization should be able to support the implementation by providing these required additional activities. On the other hand, administrative MDM issues can lead to the further questions of the data management, which are covered by Data Governance [11], another well-known domain.

Thus, every component of the model has a *software* part and an *analytical* part. For example, creating master data logical model is analytical work only, but data cleansing concerns both parts. The latter could be stated about implementing special cleansing rules, and their usage, including result analysis, leading to potential other rules, etc. Usually, MDM products and some ready-made tools are used but some steps specific to an organization can be implemented as separate scripts, this is so-called additional software created withing the MDM project.

Let us now describe the main packages of the model.

- **Data Collection.** This package includes components that identify master data candidates (raw data), and perform its further analysis and preprocessing. Furthermore, it also includes the implementation of the DS systems access.
- **Data Processing.** This package contains functionality to provide development and storage of the master data in the data hub, including creation and maintenance of a logical data model, as well as classification, hierarchy building, and so on. Therefore, the data hub receives raw data from DS that has been preprocessed by the previous package. This package processes it, transforming into the master data.

- **Data Delivery.** This package contains components that deliver master data to DC systems. Note that DC and DS systems can coincide, fully or partially. This emphasizes the issues of the distribution master data access rights and various data delivery modes. We have identified the following modes of the master data delivery: (i) package-driven, (ii) real-time, (iii) subscription-based.

It is important to note that the proposed model is focused on incremental/repeating master data updates in the data hub that take into account new raw data, instead of a single-time upload.

Packages and their functional components are presented in Fig.1.

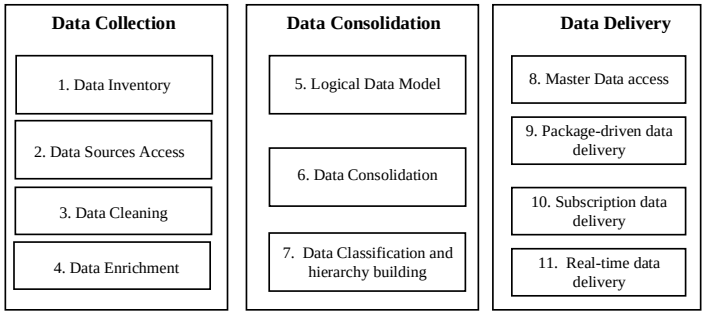

Fig. 1. A schematic representation of a typical MDM solution.

5.1 Data Inventory

This component identifies DS systems and determines which organization's data in particular is to become master data. It is important to define precisely what master data will consist of because as the variance in data increases, the complexity and costs of the MDM project increase as well. Furthermore, only those attributes that the DC systems are demanding should be collected at this stage. Accessing some data may be challenging due to its security policies, or organizational and technical access complications.

It is essential to determine data types, real values range of each attribute, their default values, etc. This package also identifies the trust level for each DS. Some DS systems may have a very low trust level, for example if their data have not been updated for a long time. Therefore, they should be accessed only in extreme cases.

This functional component is predominantly analytical.

5.2 Data Source Access

Since the raw data used to create master data resides in the DS systems of the organization, an MDM solution requires an implementation of the software-based access to this data in order to, at least, upload it into the hub. In most cases, data upload is a repeating procedure that is performed regularly by the functioning MDM solution. In order to automate it, considerable volume of technical work is required: DS systems may be outdated and may not even have external software access interfaces.

This functional component is predominantly software-based. The scope of work largely depends on the level to which internal data exchange has been established and developed within the organization. The organization may have already implemented data "transportation" layer between various information systems, for example, based on an enterprise data bus.

5.3 Data Cleaning

This component concerns error detection and potential fixing as well as normalization of the input data from different DS systems before its upload to the data hub. This step is important since duplicate detection and data consolidation would become quite challenging without it. Data cleaning is a high effort process, and 100% result is rarely achievable. In any case, it is reasonable to perform

preliminary cleaning that includes normalization, format unification of the significant attributes. This type of cleaning is not absolutely required, but it drastically simplifies further essential steps of data consolidation and entity resolution.

This component is both software and analytical. Despite of a large body of ready-made software it is often the case that either a tool needs modifications to correctly process various data formats in question, or custom data cleaning logic needs to be implemented. For example, a DS system may store several values in a single attribute, and these values have to be split into the corresponding fields, which requires specialized software utility/script development.

5.4 Data Enrichment

An organization may need to augment its master data contained in their DS systems with some open-source data. A prime example is supplementing contractual data with legal and tax-related information. This component is software and analytical since it requires analysis and annotation of the corresponding data obtained from DS, as well as implementation of the software-based access to the public sources used for data enrichment.

5.5 Logical Data Model

This component is intended for development and maintenance of the logical model of master data. The model needs to represent the structure of the consolidated data containing all the attributes collected from the DS systems of the organization. It is essential for further processing of the master data, as well as its delivery to DS systems. One of the most important steps in the logical data model creation is the reconstruction/detection of various relationships within the data that are absent in the DS systems, but to be appeared during the consolidation process.

Logical data model creation, which is identifying which entities, attributes, and relationships should appear in the new schema, is an analytical process. However, it should be supported with modelling software and visualization tools that operate on a list of attributes and entity relationships (often of different types), as well as a software link of the master data model to the corresponding mapping of its elements to DS and/or DC systems. At the same time, some aspects of these tools may need to be modified for the given project.

5.6 Data Consolidation

This component is responsible for uploading data from DS systems to the hub and consolidating it according to the previously created logical model. This process is performed automatically; however, data collisions or conflicts arise during its execution, which can be resolved in the following ways.

- "Manual" approach involves a domain or subject area expert to resolve the data conflict. It is employed when automatic conflict resolution is inappropriate: i.e., for critical data (such as legal) that cannot tolerate errors.
- Machine learning techniques, which learn typical situations in order to automatically resolve consolidation conflicts.
- Combined strategy: domain experts can definitively resolve preliminary results provided by machine learning algorithms. This approach can drastically lower the conflict resolution efforts.

Raw data can be uploaded to the hub from DS systems once, for example, in case the Centralized architecture is used, or, in case there are DS systems that discontinued their function but still contain valuable data. Otherwise, besides the initial raw data load, incremental uploading procedure should be implemented.

This component is both software and analytical. Its software part is the analyst's data consolidation workplace, which often has to be modified to support handling specialized data and implementation of custom data consolidation and conflict resolution rules that could be used automatically.

Additionally, if machine learning algorithms are used then they should be trained correspondingly, e.g. they could be used to implement updatable or self-configuring rules of data conflict resolution.

5.7 Classification and Hierarchy Building

Organizations require their master data to be as systematized and ordered as possible. Often it is necessary to perform data *classification* that means defining data clusters, and identifying their significant characteristics. At the same time, organization's data may need to be linked to the external classifications such as federal standards, industrial classifications and so on. In many cases data also requires *hierarchization*: for example, reconstruction of the hierarchy of customers or suppliers. Consider a case when a new customer of the organization belongs to the same branch of a large corporation as another one, and thus there is no need to perform new security clearance, which saves a lot of time and resources during contract preparation. Another example of the data relationship: if it is known that two customers are married, the organization can offer them additional services. Classification and hierarchy building is performed on the unstructured data, and thus it can be combined with data enrichment.

This component is analytical concerning the definition of the rules of hierarchy building and classification. It should be supported by the appropriate software that allows to "test" these rules on a small volume of data first, and only then applies them to the whole set of data. At the same time, various machine learning methods are increasingly employed as analytical recommenders and semi-automatic data classifiers. Almost everything in the toolset of this package requires modifications and configuration due to business cases and data specifics.

5.8 Access Rights Model

Different data consumers can be situated in various business segments of the organization and therefore have different data access rights. For example, some consumers should be able to access all data, but others should not see certain attributes. This component defines and implements a strategy for master data access rights. It should be based on the existing user roles withing the organization and corresponding data access rights, and it requires interacting with the security team.

This component includes tasks that are mostly analytical and not time-consuming. The policies of master data access are implemented via the means of the information system security management. However, creating a corresponding specification is a highly significant task that requires deep knowledge of the data and business processes, as well as the structure of the organization. Data security granularity and its efficient implementation are the key to the success of the MDM project.

5.9 Package-Driven Data Delivery

This component is responsible for uploading and updating master data in the DC systems according to a given schedule. Many DS systems focus on uploading master data into datamarts, which they are further handling according to their own procedures. In general, each datamart uses its own master data fragment. It is reasonable to implement a separate management tool for datamarts in order to refresh its master data, monitor and audit the DC queries. This approach allows to track not only what master data is consumed by which DC but also what conflicts arise and how they correspond to the master data consumption.

This component is mainly software including implementation and configuration of an MDM system's interface to send data to the appropriate DSs or utilize corresponding ETL procedure(s)[2]. The analytical part is not too complicated and consists of determining which DCs and master data fragments that require this particular strategy.

[2] ETL (Extract, Transform, Load) is the general procedure of extracting data from source information systems, transformation of the extracted data and its delivery to the destination systems [17]. MDM may be one of the boxes of the ETL procedure.

5.10 Subscription-Based Data Delivery

Within this mode, every DC system subscribes to a particular fragment of the master data (fragment of the logical model or a subset of entities and their attributes). Then one or several queues receive the newest version of this master data after each update. Finally, all the DC systems load their updates from the queues according to their subscriptions. This model's convenience lies in the hub being a centralized place of management and administration of the master data across all DCs. The complexity of its implementation is that either the existing queue mechanism used by DCs has to be reused, or the DCs have to be modified in order to use MDM solution queues. This component is software and analytical at the same time similar to the previous one.

5.11 Real-Time Data Delivery

This component consists of master data delivery to DCs in real time, i.e., immediately after the master data has been modified (consolidated and updated). This mode is often hard to implement due to conflicts that arise on the consumer's side as the DC system can temporarily block access to the data fragment during some operations execution, which leads to delays in the master data updates in the hub. This component is both software and analytical.

6. Case Study

In the presented case study, we consider organizations that belong to different business sectors (the energy industry, telecommunications, transportation, retail, government, and machine manufacturing industries) in order to demonstrate that the iterative approach and the proposed functional model are universal and independent of the subject area in question. The list of the organizations and their characteristics is presented below. We have considered medium-size and large business organizations (i.e. more than 10 000 employees) because the iterative approach to MDM is most relevant for them. Due to their large size, these organizations contain many various data domain segments which are often located in the separate departments. In their turn, the latter may be independent subdivisions of the whole organization having their own local information systems and data. The iterative approach enables MDM implementation not for the whole organization but for selected individual department considering limited data domain segments and based on the department business needs and scenarios. After successful completion of the first MDM project the following MDM project may be initiated in the organization for another department. Departments and their domain segments considered in this case study are presented in Table 1.

The original goals and business case(s) of the studied MDM projects (let’s code the MDM solutions of these projects in bold to reference further in this work) are described below.

- **IP:** Producing a global inventory catalog in a large corporation in the energy sector (sales department).
- **PIM:** Consolidation of technical and marketing information to produce a product catalog within a telecommunication organization (sales department).
- **PSC:** Creation of catalogs of products/services and customers/suppliers for procurement purposes, management of the relationships between customers and suppliers (supplier department).
- **CS:** Enrichment of a clothing retailer’s customer database with social media data in order to identify and support influencers (customer support department).
- **CPO:** Personal data consolidation across regional and federal databases – real estate, vehicles, etc. (citizen communication department).
- **CDV:** Identification and verification of the customer data for a global corporation; management of legal and related hierarchies. Support of the customer’s segmentation (customer onboarding department).

To meet these requirements the MDM projects were conducted within the corresponding department of the organizations, which characteristics are described in Table 2.

Table 1. Studied Organizations.

Organization code name	C1	C2	C3	C4	C5	C6
Business sector	Energy	Telecom	Transport	Retail	Government	Energy & machine manufacturing
Organization size (number of employees)	48 800	36 000	729 000	14 630	100 000	184 000
Organization department involved	Sales	Sales	Supplier	Customer support	Citizen communication	Customer onboarding
MDM domain segments selected	Inventory	Products, personal data	Products, services	Products, personal data	Personal data	Customer data

Table 2. Real-life MDM projects.

№	**Project code**	**Org code**	**Overall data volume** (K records)	**Master data volume** (K records)	**Number of data sources**	**Project sizing** (man-months)	**Project duration** (months)	**Project work distribution, %**		
								sw	**analysis**	**dp**
1.	IP	C1	500	350	15	9450	18	23	70	**7**
2.	PIM	C2	100 000	70 000	12	3024	12	30	30	**40**
3.	PSC	C3	2 000	700	4	7560	24	30	45	**25**
4.	CS	C4	70	30	5	1890	9	16	64	**20**
5.	CPO	C5	170 000	30 000	15	4536	12	45	25	**30**
6.	CDV	C6	4 000	1 500	213	10080	24	56	24	**20**

- “Overall data volume” indicates the overall data volume in these segments measured in thousands of records.
- “Master data volume” presents the resulting master data volume, extracted and consolidated in the context of the studied MDM projects. It is obvious that not all of the domain segments’ data should be included in the master dataset, i.e. be considered as master data. The volume of these data is also measured in thousands of records.
- “Number of data sources” describes the number of data sources for every MDM project. This is one of the primary characteristics for determining whether the MDM approach is suitable to address the needs of an organization as MDM is focused on consolidating data from various sources. If, however, there is a small number of sources, consolidation is unnecessary and therefore, other means should be used to satisfy the organization needs.
- “Project sizing” indicates the overall estimation of the MDM projects (in man-months).
- “Project duration” shows the calendar duration of these projects (in months).
- “Project work distribution” shows percentage distribution of the total work between software development (sw), analytical work (analysis), and actual data processing (dp) including all the uploads, incremental loads, etc.

Note, that MDM projects are actually smaller than projects of information system development from the scratch. Moreover, as we can see in, the ratio of software development in these projects is 33% on average, analytical work is the most significant – 43% on average.

It should be noted that the authors of the article have actively participated in these projects, utilizing the proposed functional model for defining the key aspects of MDM projects. Table 3 shows which components of the model were implemented in the specified MDM projects. The following scale was used:

- *High* denotes that the component is crucial for the project (business-critical or technologically complex)
- *Med(ium)* refers to a component that is necessary, but was not high-priority or effort-demanding
- *Low* denotes a component that has a "light" version: it either existed within the organization previously and required only modifications within the given MDM project, or was moved out into a separate project
- *N/A* means that the component was not required for the project

Let us consider the data presented in Table 3 from the MDM project's perspective.

- **IP:** One of the project's business requirements was to automate complex organization data handling policies that were employed in more than ten various departments. Implementing a material resource classifier (*data classification and hierarchy building*) as well as creating a *logical data model* were its key points.
- **PIM:** The project was mainly focused on the *data inventory* of the organization products taken from various DSs, creating a unified *logical master data model,* and *data consolidation*. Besides, it required the construction of a product tree that contained all product information, including financials, so the sales department and financial experts would be able to perform further analysis (*data classification and hierarchy building*).
- **PSC:** The project addressed *consolidation of data* of the goods purchased by the organization. It was necessary to combine all the information about purchased items obtained from various commodity nomenclatures, and to create a list of contractor services. Besides data consolidation, the project was also focused on the *access rights model* and *subscription-based data delivery*.
- **CS:** The project required *data enrichment* and *consolidation* with the purpose of identifying top influencers on social media among the customers of the clothing retail organization to provide them with some special treatment and preferences.
- **CPO:** This project was developed for a government management service in order to create a "smart" personal account of a citizen. It was necessary to integrate the account within federal and regional information systems. The particular goals of the project were information security and corresponding *access rights model*, as well as *real-time* and *subscription-based data delivery.*
- **CDV:** The organization her had hundreds of thousands of customers worldwide, and so its customer onboarding procedure was very complicated. Before the MDM project got implemented, the procedure had taken 21 days, but after it got moved to production, the duration decreased to just 8 days. This project was focused on *data inventory* and *logical master data model,* which facilitated duplicate detection of legal entities and the search of their affiliated entities. Besides, it's required *real-time master data delivery* in order to speed up the target business process.

So, we can conclude that the proposed MDM functional model objectively describes organization's needs and allows to consider MDM projects significantly decreasing the costs and risks of their implementation. Additionally, the proposed model appears to be convenient to identify high level functional components of these projects. Besides, it can be seen that different MDM projects have significantly different focal points, and many model components are marked as "low" or "N/A", i.e., they were very easy or omitted altogether, which also has cut implementation costs.

Table 3. MDM Solution functionality.

Functional Model Components	Target MDM-solutions					
	IP	**PIM**	**PSC**	**CS**	**CPO**	**CDV**
I. Data Collection						
Data Inventory	Med	High	Med	N/A	Med	High
Data Source Access	Low	Med	Low	Low	Low	Med
Data Cleaning	Med	Low	Med	Med	Med	Med
Data Enrichment	N/A	Med	Med	High	N/A	N/A
II. Data Processing						
Logical Data Model	High	High	Med	Low	Med	High
Data Consolidation	Med	High	High	High	Med	Med
Data Classification and Hierarchy Building	High	High	Med	N/A	N/A	Med
III. Data Delivery						
Access Rights Model	Low	Low	High	Low	High	Med
Subscription-Based Data Delivery	Med	Low	High	Low	High	Med
Real-Time Data Delivery	Med	N/A	Med	Med	High	High
Package-Driven Data Delivery	Med	Med	Med	N/A	Low	Med

7.Related Work

In recent years, a significant number of standards and data management methodologies has emerged: DAMA-DMBOK [3], CMMI Data Management Maturity Model (DMM) [18], IBM Data Governance Council Maturity Model [19]. MDM is included in these standards, and DAMA-DMBOK considers it most extensively.

The following are the DAMA-DMBOK models that concern master data management.

- "Context Diagram: Reference and Master Data". This model defines the intents of data management, describes all the necessary activities (see the following model), their inputs and outputs, identifies suppliers, participants, and consumers, introduces the notion of a business driver and a technical driver, and finally, defines the necessary methods, tools, and metrics.
- Model that describes activities for MDM implementation: (i) identifying drivers and requirements of MDM, (ii) analyzing and evaluating data sources, (iii) selecting data hub architecture, (iv) modeling master data, (v) implementing master data management and integration; (vi) defining control policies and ensuring compliance.

"Context Diagram: Reference and Master Data" is a comprehensive vision of the MDM framework. Nevertheless, it is predominantly a descriptive model that provides a conceptual framework rather than a tool suitable for a practical use.

Let us consider in detail the model of MDM implementation activities within an organization. Despite its rather pragmatic viewpoint, it omits the issues concerning software. In general, the vision behind this model is based on analysis and administrative work, while we substantially highlight the role that software plays in MDM implementation.

Another type of existing studies is focused on MDM software in particular. Generally speaking, these studies consider MDM product adoption methodologies which every MDM manufacturer possesses: a good example would be the Velocity Methodology by Informatica [20]. These methodologies are focused on adoption of specific products of the selected manufacturer to the organizations. In contrast, the model proposed in the current paper is independent from any particular MDM product.

Among the numerous papers by Gartner who widely analyze questions of MDM implementation, [21] can be highlighted. It proposes a framework consisting of seven components: vision, strategy, metrics, information governance, organization and roles, information life cycle, and enabling

infrastructure. Similarly to our model, it is intended for the early stages of MDM implementation. However, the scope of this model is the conceptual MDM implementation within an organization, i.e., it assumes a top-down strategy. In contrast, our model is intended for the iterative strategy focused on satisfying specific organization needs.

An example of an academic MDM research is presented in [22], which proposes a model for analysis of the master data life cycle in a organization. It is intended for analyzing the existing master data management cycle, i.e., identifying missing activities and bringing to light its challenges. The main components of this model are: (i) data portfolio (organization data that undergoes master data management); (ii) data and system design (data hub architecture, IT-related aspects of MDM) (iii) data management (development and maintenance of the master data (including its updates), quality control); (iv) data maintenance (monitoring and controlling the MDM process). Note that the proposed model does not consider the software aspects of MDM solutions. Furthermore, it is not focused on the specific MDM implementation tasks within an organization.

8. Conclusion

In the current paper, we propose a functional model of an MDM solution intended for development within the iterative strategy. The model is designed for the early stages of the MDM implementation, and its main purpose is to translate the needs of an organization into the MDM terminology in order to estimate the percentage of MDM specific tasks. It allows to plan and estimate necessary functionality of the MDM solution and proceed with the tools selection and composing technical requirements. We provide a case study of the real MDM projects that were implemented with the model in question.

As a further direction of our research, we plan to develop evaluation techniques for the early-stage MDM project functionality and fine-grained metrics of the MDM solution complexity. Additionally, we intend to map the functionality of a typical MDM solution to various MDM products, as well as integrate our approach more tightly with knowledge management [23–24] and visualization techniques [25–26].

References

[1]. Khatri V., Brown C.V. Designing data governance. Communications of the ACM, 53(1), 2010, pp. 148– 152.
[2]. DAMA-DMBOK: Data Management Body of Knowledge, 2017. 588 p.
[3]. Andrichenko A.N. Tendencies and condition in the field of reference data management in the engineering industry. Ontology of designing, 2 (4), 2012, pp. 25–35. (in Russian).
[4]. Nemtsov E.F. Reference data in the intelligent railway transport management system. Automation, communication and Informatic, 2, 2020, pp. 15–18. (in Russian).
[5]. Golubev S.S., Lotsmanov A.N., Kuzin A. Y., Soloviev V.G., Kozlov A.D., Grigoriev B.A. The branch system of the National Standard Reference Data Service for the Oil and Gas Complex. Legislative and applied metrology, 3, 2020, pp. 12–16. (in Russian).
[6]. Yanchenko G.A. On the standard reference data density properties of rocks. Mining information and analytical bulletin, 8, 2011, pp. 111–115. (in Russian).
[7]. Chigirinsky Y., L. Method of reliability improvement of reference data. Izvestia Volgogradskogo universitets, 13 (86), 2011, pp. 55–61. (in Russian).
[8]. Gartner Glossary, https://www.gartner.com/en/glossary, accessed 30.02.2025.
[9]. Walker S. Dayley A., Parker S., Hawker M. Magic Quadrant for Master Data Management. Gartner, 2021.
[10]. Zmud R.W. An Examination of ‘Push-Pull’ Theory Applied to Process Innovation in Knowledge Work. Management Science, 30 (6), 1984, pp. 727–738.
[11]. Ladley J. Data Governance: How to Design, Deploy, and Sustain an Effective Data Governance Program. Academic Press; 2nd edition, 2019. 350 p.
[12]. Kuznetsov S.V., Koznov D.V. Master data management in an iterative approach. Ontology of designing, 11(2), 2021, pp. 170–184. (in Russian).
[13]. Silvola R., Jääskelainen O., Kropsu-Vehkaperä H., Haapasalo H. Managing one master data – Challenges and preconditions. Industrial Management & Data Systems, 111(1), 2011, pp. 146–162.

[14]. Yourdon E., Constantine L.L. Structured Design: Fundamentals of a Discipline of Program and Systems Design. Yourdon Press, 1975. 348 p.
[15]. Jacobson I. Object-Oriented Software Engineering. ASM Press, 1992. 528 p.
[16]. Ould M.A. Business Processes: Modelling and Analysis for Re-Engineering and Improvement. Wiley, 1995.
[17]. Vassiliadis P., Simitsis A., Skiadopoulos S. Conceptual modeling for ETL processes. DOLAP, 2002, pp. 14–21.
[18]. CMMI Data Management Maturity Model (DMM). CMMI Institute (website). http://bit.ly/1Vev9xx, accessed 30.01.2025.
[19]. IBM Data Governance Council Maturity Model https://ibm.co/2sRfBIn, accessed 30.01.2025.
[20]. Velocity Methodology. Best Practices. Informatica. 2008.
[21]. O'Kane B., Moran M. P. The Seven Building Blocks of MDM: A Framework for Success. August 2016. Gartner. ID: G00311161.
[22]. Ofner M.H., Straub K., Otto B., Österle H. Management of the master data lifecycle: a framework for analysis. J. Enterp. Inf. Manag, 26(4), 2013, pp. 472–491.
[23]. Gavrilova T.A., Kudryvtsiv D. V. Knowledge management: from words to business. Intelligent Enterprise: RE, 12–13 (101), 2004, pp. 48. (in Russian).
[24]. Gavrilova T.A. Knowledge presentation in expert system АВТАНТЕСТ. Izvestia Acadimii Nauk USSR, 5, 1984, pp. 165–173. (in Russian).
[25]. Koznov D.V., Peregudov A.F., et al. Visual environment for designing broadcast software. Sistemnoye programmirovaniye, 2 (1), 2006, pp. 142–168. (in Russian).
[26]. Ivanov A., Koznov D., et al. Behavior model RTST++. Zapiski seminara kafedry sistemnogo programmirovaniya "Case-tool RTST++", 1, 1998, pp. 37–52. (in Russian).

Информация об авторах / Information about authors

Сергей Викторович КУЗНЕЦОВ – преподаватель кафедры прикладной кибернетики математико-механического факультета СПбГУ, исполнительный директор ООО «Юнидата». Сфера научных интересов: управление данными, управление мастер-данными, алгоритмы на графах, глубокое машинное обучение.

Sergey Viktorovich KUZNETSOV – a lecture of St.Petersburg University (Applied Cybernetics Chair), CEO of Unidata Ltd since 2014. Research interests: data government, master data management, graph algorithms, deep learning.

Дмитрий Владимирович КОЗНОВ, доктор технических наук, профессор кафедры системного программирования математико-механического факультета СПбГУ. Научные интересы: программная инженерия, модельно-ориентированная разработка программного обеспечения и dsls (в частности, для enterprise-приложений и телекоммуникаций), машинное обучение и нейросети в программной инженерии, техническая документация.

Dmitry Vladimirovich KOZNOV – Dr. Sci. (Tech.), Professor of the System Programming Department of St. Petersburg State University. Research interests: software engineering, model-driven software development & dsl (in particular, for enterprise applications and telecom), machine learning and neural networks for software data, technical documentation.

Дмитрий Вадимович ЛУЦИВ – кандидат физико-математических наук, доцент кафедры системного программирования Санкт-Петербургского государственного университета. Область научных интересов: программная инженерия, анализ данных программных проектов, анализ документации, системное программирование.

Dmitry Vadimovich LUCIV – PhD in computer science, associate professor of System Programming Department at Saint Petersburg State University, Russia. Research interests: software engineering, software data analysis, documentation analysis, systems programming.