POLITECNICO
MILANO 1863

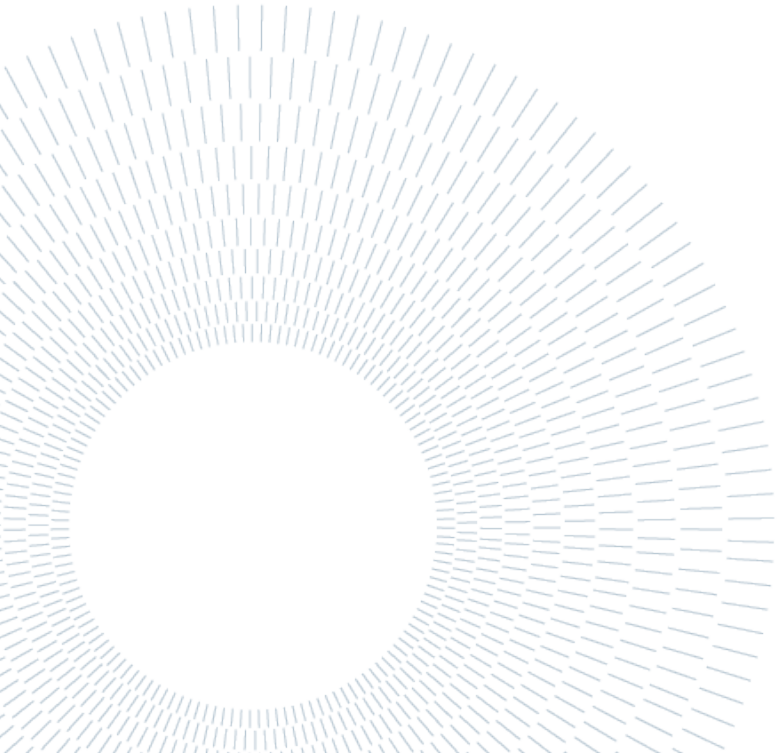
SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

A Metadata Model for Data Lake in Industry 4.0: the MADE Experience

TESI DI LAUREA MAGISTRALE IN
MANAGEMENT ENGINEERING
INGEGNERIA GESTIONALE

Authors: **Filippo Tubino, Edoardo Tonetti**

Student ID: 10568895, 10600017
Advisor: Pierluigi Plebani
Co-advisor: Cappiello Cinzia
Academic Year: 2021-2022



Abstract

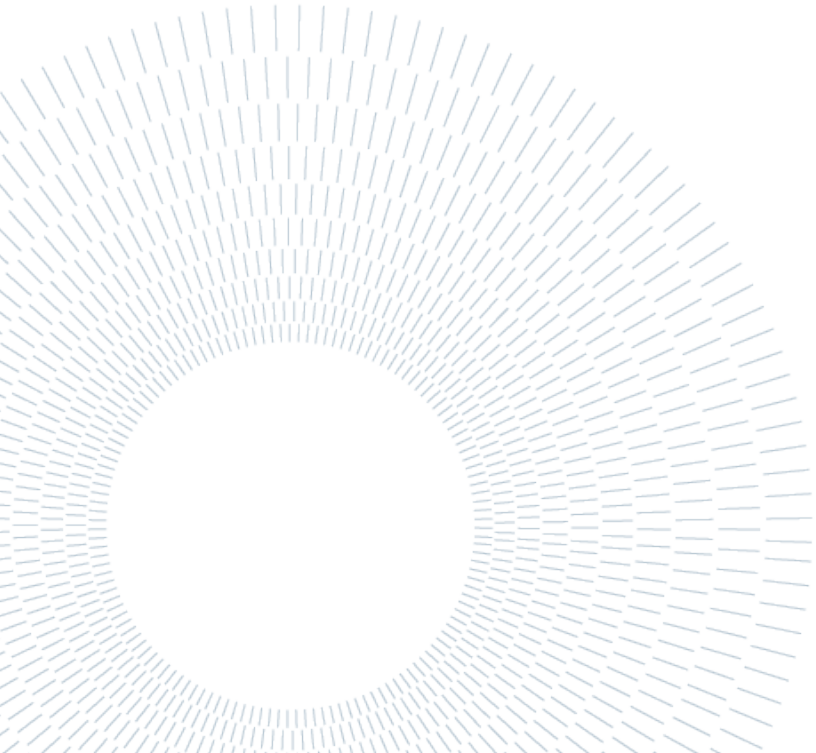
The increasing amount of data collected and used for analysis requires a change from traditional data warehouses. Data lakes are an increasingly used solution by companies to store and search for data they collect. Despite this, data lakes are still a relatively new technology and a defined approach for their implementation is lacking. Crucial to the management of this system is the management of metadata, through which data can be easily found in the repository once saved. Several researchers have proposed metadata models: frameworks for metadata management that offer different features more or less useful depending on the context of use. In this dissertation, we analyze the usefulness of different metadata categories based on the needs of MADE, the I4.0 competence center at the Polytechnic of Milan. The information obtained from the competence center area managers is analyzed to gain knowledge regarding the data lake features required by I4.0 and IoT related companies. This will allow to select the most tailored metadata model that prevents the entire system from becoming a "data swamp", a repository of data in which data analysts cannot find what is of interest.

Key-words: Data Lake, Metadata, Industry 4.0, Metadata model.

Abstract in italiano

L'aumentare della mole di dati raccolti e utilizzati per le analisi richiede un cambiamento dei tradizionali data warehouse. I data lake rappresentano una soluzione sempre più utilizzata dalle aziende per salvare e ricercare i dati da loro raccolti. Nonostante questo, i data lake sono ancora una tecnologia relativamente nuova e manca un approccio ben definito per la loro implementazione. Per la gestione di questi sistemi è di fondamentale importanza la gestione dei metadati, grazie ai quali è possibile ritrovare facilmente i dati nel repository una volta salvati. Diversi ricercatori hanno proposto dei metadata model: framework per la gestione dei matadati che offrono diverse funzionalità più o meno utili in base al contesto di utilizzo. In questa dissertation si analizza l'utilità di diverse categorie di metadati in base alle esigenze del MADE, centro competenze I4.0 del Politecnico di Milano e più in generale per le aziende legate all'I4.0. Le informazioni ottenute dai responsabili di area del MADE verranno poi analizzate per ottenere conoscenza riguardo le funzionalità dei data lake richieste dalle aziende che investono in tecnologie I4.0 e IoT. Questo ci permetterà di scegliere un modello di metadati che impedisca all'intero sistema di diventare un "data swamp", una repository di dati nella quale non si riesce a cercare ciò che è d'interesse.

Parole chiave: Data Lake, Metadati, Industria 4.0, Modello gestione metadati.



Contents

Introduction

The starting point for implementing a data lake in any organization is to understand how to manage the data within it. To do this, metadata plays a key role in organizing the data and making it readily available for future analysis. Indeed, without effective metadata management, the whole data lake risks turning into a so-called "data swamp". This term refers to a repository of data that you cannot access and in which it is not possible to find what you are looking for. All this could make data analysis ineffective and difficult, if not impossible, to conduct research to extract value from data. Therefore, the selection of the most tailored approach to metadata management is of paramount importance.

Several researchers in the literature have proposed different metadata models for organizing and managing data within data lakes. These metadata models are frameworks for the conceptual organization of data by exploiting metadata. Each of these differs in how they function and the features they enable for system management. Companies in different industries require different data lake features based on the data they collect and the analysis they perform. The existing literature lacks information regarding the requirements and needs of companies to conduct analysis in different sectors. The purpose of this dissertation is to analyze the requirements of I4.0 and IoT-related companies to select the most effective metadata model in this context of use. To do this, analyses will be conducted with MADE, the Industry 4.0 competence center of the Polytechnic of Milan, to understand how and what analyses are done in this context. This use case represents an ideal analysis environment for extracting the needed information, given the multipurpose nature of the activities that are carried out within it.

The dissertation work begins with a review of the existing literature regarding data lakes. Since a detailed analysis of the data lake requirements in this field is lacking, it will then be necessary to interview the MADE area managers to extrapolate information about data in their fields. The interviews must be aimed at obtaining information regarding the data collected, the analyses performed, and the different metadata used. In this way, it is possible to define the usefulness of different metadata categories in I4.0 environments. Metadata serves as input to enable the various functions of the data lake. So, it is possible to understand which metadata enables certain features. In this way, it is possible to identify a set of features that can be considered essential in order not to turn the data lake into a "data swamp". Once it is clear which features are considered essential for I4.0 and IoT related companies, it will be possible to select the most appropriate metadata model for organizations operating in this context. As a result, it is possible to define what are the bases for implementing a data lake and which metadata management framework is most appropriate in I4.0 environments.

1 Literature review and state of art

1.1. Data Warehouse and Big Data

Several large companies have a data warehouse in which to store all the important information for analysis aimed at decision making. The DW serves the company's management to monitor the progress of operations as well as to make decisions regarding the company's future: some examples of data use may be customer relationship management systems, data for inventory, and data on sales and purchasing

1.1.1. Big data “4V”

But with the advent of big data these traditional DWs are proving to be less and less flexible and suitable for handling this enormous amount of data. at this point a premise about big data is necessary. To summarize and explain the concept of big data and the problems that traditional DW have in relation to it, the explanation of the 4 main properties of big data i.e. the "4vs" is presented below. The first is volume, that is related to the significant amount of data to be processed. The goal is to manage this amount, starting from a lot of disconnected and erroneous data, to create information through analysis to create value throughout the company. This means cleaning and filtering basing on the correctness and usefulness of the data itself. In a traditional database, this process requires a lot of effort and often too inability to perform real-time analysis before the data is cleaned and filtered. The second v is Velocity, thus the peculiarity of Big Data solutions to manage data streams that continuously produce data. Sometimes the amount of data coming in is so large that it is not possible to process all of it, because the computational capacity is not sufficient. If the analyses to

be done are in batches, you can store the excess data to process later, but if the analyses are required in realtime or with very short timelines, then it becomes critical to be able to take advantage of the data as soon as possible. In Industry 4.0, sensors send data to the database at very high cadences, which makes velocity critical for big data especially when it comes to quality control or machinery maintenance. Then we have the third "v" the variety, that is, all the possible formats that data can have including text files, videos, spreadsheets, images, tweets and so on. For example, a company that collects data from sensors from different sources, that come in different formats, will then have to integrate the various data before storing it in the traditional database and then make the resulting analyses possible. In an environment where the number of data sources continues to grow and data formats are increasingly heterogeneous, if we think of data collected from social media, for example, it becomes increasingly critical to solve problems related to this third "v." The fourth “v” is veracity: basically this one considers all the errors and false data that might come from social media or other sources, that can mislead the analysis.

In conclusion, it is being realized that traditional data warehouses are no longer able to meet the new requirements regarding big data the numerosity, variety and high degree of performance that is required nowadays.

Anyway, even with the adoption of the architectures shown before, some problems remain unsolved for the traditional data warehouses, like the computational power that is not enough for matching data velocity, or the time wasted for integrating heterogeneous formats data. So, one of the latest trends is the adoption of Apache Hadoop software libraries building data lakes.

1.1.2. The Rise of Data Lakes

Companies are starting to extract and put data into a Hadoop-based repository without first transforming the data as they would do with a traditional DW. It's all

about storing data in a Hadoop repository without performing data cleaning or transformation operations.

So, data analysts can access the data repository whenever they want and basing on the analysis to be conducted, choose what portion to work on without a priori data preparation. So, each framework for analysis is different from the other. It's all about temporally shifting the data processing and cleaning processes, focusing only on the portion of giving you useful to creating value for the entire enterprise. Costs and time, by doing this, are greatly reduced. Therefore, in many companies there is a shift to the use of data lakes.

1.2. Data Lakes vs. Data Warehouses

A data lake is a central place to store all data, regardless of the source from which it is extracted or the format. The main advantage compared to traditional DWs is that there are no limits to the types and formats of data that can be leveraged.

All types of structured and unstructured data, from CRM data (CRM or Customer Relationship Management is a tool for managing the relationships and interactions a company has with potential and existing customers) to social media posts, videos, images and text files can be stored in a data lake without having to worry in advance about the process of cleaning and correcting the data on hand.

It is only necessary to save raw data, including errors, duplicates, and any apparently unnecessary data. Later they can then be refined as you have an idea of what to analyze. There are no limits to how data can be used for analysis purposes. In conducting the analyses, it will be possible to use certain methods and tools to understand the meaning of the data and the relationships between them. Unified access is achieved, with no more division into silos, with an overview and democratized view of the data present within the entire organization. Data captured on social media and by Internet of Things sensors, on the one hand are a great

opportunity for companies, but on the other hand they encounter problems related to volume velocity and variety.

Thus, the concept of a data lake presents itself as a solution to the diverse problems of big data. Data lake gives an integrated presentation of the data without a priori predefined schema, as it will be developed on read. In case a data schema is missing, an efficient metadata system is the key to querying the data and avoiding the creation of a data swamp, i.e., in a useless data lake. Table 1 resumes the main differences between data warehouses and data lakes.

Table 1 Enterprise Data warehouse vs Data Lake

Attribute	EDW	Data Lake
Schema	Schema-on-write	Schema-on-read
Scale	Scales to large volumes at moderate cost	Scales to huge volumes at low cost
Access Methods	Accessed through standardized SQL and BI tools	Accessed to SQL-like system, programs created by developers and other methods
Workload	Support batch processing, as well as thousands of concurrent users performing interactive analytics	Supports batch processing, plus an improved capability over EDWs to support interactive queries from users
Data	Cleansed	Raw
Complexity	Complex Integration	Complex processing
Cost/Efficiency	Efficiently uses CPU/IO	Efficiently uses storage and processing capabilities at very low cost
Benefits	▪ Transform once, use many ▪ Clean, safe, secure data ▪ Provides a single enterprise-wide view of	▪ Transforms the economics of storing large amounts of data

	data from multiple sources ▪ Easy to consume data ▪ High concurrency ▪ Consistent performance ▪ Fast response times	▪ Supports Pig and HiveQL and other high-level programming frameworks ▪ Scales to execute on tens of thousands of servers ▪ Allows use of any tool ▪ Enables analysis to begin as soon as the data arrives ▪ Allows usage of structured and unstructured content from a single store ▪ Supports agile modelling by allowing users to change models, applications and queries

1.2.1.1. Schema-on-write vs. Schema-on-read

The first difference that is shown in the figure is that between schema-on-read and schema-on-write: traditionally when you perform analysis on a dataset you have a clear objective. For example, in data warehouses you consider the dimensions of analysis related to a fact with specific measures for each dimension and sub-dimension (e.g. if the fact is the sale of a product, the dimensions will be time, place and revenue). The data stored are varied and divided into hierarchies, so there is a possibility that data, irrelevant to the analysis, will be collected. It becomes clear that if the database is not too large, it will be possible to carry out the analyses directly with traditional databases, but if we are talking about big data, problems of complexity emerge if we are talking about arranging the dataset according to relationships, dimensions, and hierarchies in order to carry out query operations.

Hence the difference between schema-on-read and schema-on-write:

1.2.1.2. Schema-on-write

To be analyzed in DW, data needs a definite structure, since without it, it is impossible to perform operations on the dataset according to a schema that highlights data attributes, dimensions, and relationships between various subjects. In a DW data scientist already has a priori in mind the way in which the data will be stored according to the structure of dimensions and hierarchies proper to each dataset. So, it is schema-on-write since he already has the structure that the data will take in the dataset before the data is entered into the data management system. This can be constraining in the analysis phase, because a portion of data may be missing to complete information of a certain type, but more importantly some portions of the dataset may be useless for current analysis purposes and so it has been therefore unnecessary to waste time in arranging a large amount of data, such as big data, following a rigid and very precise structure. On the other hand, each time we conduct a new analysis, we focus only on the portion of the dataset that interests us at that moment, sorting out each time all and only the data that we find to be useful at that moment. In this case, we would not waste time a priori and it is possible to conduct analyses with complete datasets. For example, if we are only interested in the sales of a particular region, the data analyst will only take data related to this.

1.2.1.3. Schema-on-read

So, in data lakes we first load all the raw data available without respecting any structure: no preliminary analysis or processing is done to arrange the dataset according to a pattern, it is ingested and stored. There is only a data catalog that will serve those who are interfacing with the data lake to understand its general meaning. Schema-on-read since we organize the data and choose the analysis target on a case-by-case basis, depending on what we need on each occasion. The schema is decided on a case-by-case basis: which data we need and which structure to use for it is the first thing that is defined. Then, in order to work on the collected dataset, it is necessary to perform duplicate recognition work, cleaning and eliminating spurious data. It is even

important to annotate additional information that enriches the dataset by making the individual data more understandable and recognizable. As we will see later the data of the dataset i.e., metadata are of fundamental importance for the management of a data lake, the annotations priorly cited can therefore be done either in the cleaning phase or even before. Many times the usefulness of a data lake comes through the metadata that allows us to query and recognize specific data within the entire lake. Some examples of useful metadata are author, provenance, changes made, tags, and associations with other lake elements. So, metadata is of paramount importance for the governance layer of data lakes.

1.2.1.4. Costs

Another difference with DWs is that in data lakes the performances results faster using low-cost storage, scaling in this way to higher volumes containing the costs. This is due to scale economies since all the data needed to companies are just kept as copies uploading them into Hadoop. According to users, data lake cost is around $1.000 per Terabyte.

1.2.1.5. Scalability

As anticipated earlier, traditional DWs suffer in the face of the "4vs" inherent in Big Data, due to the lack of flexibility given by the rigid structures with which the data are organized, they do not scale to the volumes of data at which data lakes arrive. While data lakes are cost- and time-efficient in both storage and data usage, DWs limit companies in using all their data, since there is the loss of time related to dataset preparation.

1.2.1.6. Accessibility

In addition, data lakes have the characteristic of facilitating accessibility and integration compared to DWs. Because they are schema-on-read, you do not have the rigid silo structure of traditional data warehouses. It is possible to postpone the dataset

structuring until the time of analysis in future. In doing so, one can also enrich analysis by using data that otherwise would not have been considered.

1.2.1.7. Complexity

The complexity in traditional data warehouses lies precisely in creating the structure of the dataset in a schema-on-write manner, then integrating all the data collected, cleaning it, and thus making it usable for data users. In fact, when we talk about preparing big data for use, we often talk about ETL (extraction, transaction and loading) phases that take longer than analysis. In a data lake, data can be analyzed efficiently by these new paradigm tools without too much preparation work. Data integration requires fewer steps because data lakes do not impose a rigid metadata schema. Schema-on-read allows users to create custom schemas in their queries at run time. In contrast in data lakes the so-called preparation phase takes much less time and steps. There is no rigid schema, but rather ad-hoc schemas built for each analysis that will be done in future. Thus, preparation is much less labor-intensive. Conversely, the complexity shifts to the data processing phase, which is entrusted to data scientists, data developers and business analysts only when the data has been curated.

1.3. The Implementation of Data Lakes into Enterprises

After the advent of data lakes, IT managers in companies began to wonder if it was possible to make data lakes and data warehouses coexist and how to make the most of this new technology. The idea is to unify the positive aspects of one and the other, to have a larger database that can be queried more efficiently. As a first step, you want to add the data lake to the data warehouse for making the dataset more consistent. The goal is to have only a data lake where no more silos or clusters of the data warehouse can be distinguished and where it will be possible to query the dataset through programming. At each analysis, the dataset composed by the union of DWs and data lake will be queried to extract information. Data lakes are increasingly gaining a

foothold in the global landscape, and almost all data lake companies are using Hadoop.

1.3.1. Problems Solved

With data lakes, companies aim to solve two problems. The high costs due to the rigidity of data structures in data warehouses makes the ingestion process laborious and time consuming. The siloed view is therefore replaced by a repository without a predefined schema in which data of different formats come from heterogeneous sources and can be saved. The solution represented by data lakes allows to increase accessibility and reduced complexity in the ingestion phase.

The second problem, on the other hand, is more conceptual: with big data, the velocity of data is so high that very often it is impossible to know anything about the data that are arriving. In the case of DWs, this would be a problem, since only structured data and information can be accepted, thus losing a lot of data that could be useful during analysis.

1.3.2. The Implementation

In every company, therefore, the path to arrive at data lake implementation is different. It depends on the corporate culture, the thinking of IT managers, the level of maturity of as-is technology, and so on. The following steps represent best practices for implementing a data lake:

The following will list the steps for introducing a data lake into an enterprise.

- **Step 1:** before thinking about the data lake itself, one must focus on how to populate it. At this stage it is essential to get as much data as you can from new sources. It is also critically important to learn how to handle Hadoop even when doing very trivial analyses.
- **Step 2:** select the tools most useful to your field of analysis and learn how to use them.

- **Step 3: t**hen proceed by trying to encourage coexistence between DW and data lake by sharing their respective content with each other.
- **Step 4:** in this final step, you need to add all business functionality to the data lake to complete its installation. So, data lakes are increasingly gaining a foothold within the global landscape, especially in larger companies.

1.3.3. Data Lake Users

For obvious reasons data lakes cannot be queried with SQL, just like databases, consumers are not aware of the methods used to collect and upload contextual data. Hypothetically, users are supposed of understanding how to easily combine and integrate data from various data sources without any prior training or experience. Moreover, regardless of structure and schema, data users are assumed to be aware of incomplete datasets.

All the characteristics listed above will never be verified for a simple business analyst who is not comfortable with programming languages. Therefore, it will be knowledgeable users who will interface with the data lake, such as data scientists and IT experts. The way in which companies interface with datasets will therefore change, as the relationship between IT and business changes radically. Increasingly important, therefore, will be the members of the IT department who will be the real users of the new data lakes.

1.4. Industry 4.0 and Data Lakes

This chapter will discuss the adoption of data lakes in the Industry 4.0 domain. The benefits related to the use of this new technology as an enabler for efficient production management and beyond will be listed. Companies are increasingly beginning to realize the potential of the data lake especially relative to joint use with other technology trends that are gradually digitizing and changing the way business is done.

1.4.1. What is Industry 4.0?

As defined by McKinsey (2015) Industry 4.0 refers to "the digitization of manufacturing, with sensors embedded in almost all product components and production equipment, ubiquitous cyber-physical systems, and analysis of all relevant data." It is driven by four groups of disruptive technologies: data, machine-human dualism, data analytics, and physical-digital dualism. Data is the fuel of every company. It has become of paramount importance in terms of management, development, and operations in every company. Therefore, the management and subsequent use of data is a critical success factor for Industry 4.0. The second driver is machine-human dualism, the interaction of which enables the enrichment of the worker's work by departing from pure automation and making room for smart exoskeletons or augmented reality. The third is obviously data analytics, then the transformation of data collected by IoT tools into useful information for the entire shop floor. The fourth driver, on the other hand, is the physical-digital dualism; tools such as 3D printing, robotics or image sensing are proof of this.

One of the most important applications of industry 4.0 is product development. Thanks to the huge amount of data gathered by sensors and IoT devices, designers and developers can have access to a large set of information, that will allow them to significantly improve the goods produced. Product developers today operate in a dynamic and digital age where data on technological goods and components can be gathered and used. Industry 4.0 is related to intelligent items that can communicate and offer internet-based services. In order to provide customers with a significant additional benefit, physical hardware components are therefore coupled with software. The amount of data and information increases considerably when these new goods are developed, but fresh developments can be made using data, designs, and solutions already in existence.

1.4.2. Driver of Industry 4.0 and Data Lakes

All the benefits of Industry 4.0 such as business process improvement, product development, maintenance planning, and so on, are closely linked to efficient data collection, management, and analysis. But what is the basis of data analysis? Obviously, before focusing on analysis, the data itself must be collected and stored. This is where data lakes can play a key role in enabling the purposes of Industry 4.0.

The main benefits of using data lakes to enable Industry 4.0 will be discussed below.

There is no need to organize, process, or filter the collected data and moreover, unstructured, semi-structured, structured, relational and other types of data can all be captured using data lakes because they do not have a predefined schema for data organization.

Additionally, all of these data can be gathered at the same time even if coming from a variety of sources, sensors, IoT devices, machines, operators or external sources. They are being kept in one location, unprocessed, unorganized, and unfiltered in their original, raw state. The gathered sets, however, can be altered without running the danger of compromising the data storage because Data Lake does not allow any set structure or schema.

In addition, another advantage concerns the extraction of data for the analysis phase. In fact, if a portion of data is taken out to conduct analyses, even if the data is modified, it can be still recoverable to its original state, since a copy of the data is automatically saved in the repository. Thus, accessibility to the raw data is never lost even after changes are made and analyses are conducted.

So, if a manager needs data related to any operation that takes place on the shop floor, he or she can go into the lake and retrieve what he or she needs, since raw copies will still be washed out in the cloud. By doing so, data in the lake will always remain in its original state of collection to therefore allow flexible and ad-hoc analysis for future users.

1.4.3. Democratization

Data democratization is another benefit of using data lakes to store data in Industry 4.0 related companies. The data, in fact, is for all intents and purposes accessible to all members of the company. So, we are talking about supervisors, managers, operators, suppliers, and customers if necessary, who can stay up-to-date on everything that is happening within the company via the data lake. Corporate communication is greatly facilitated, thus eliminating time for communications made unnecessary by the implementation of the data lake.

Thus, it will no longer be necessary for someone to prepare data reports or to present them in a summary manner. Anyone who is interested in learning about the data will be able to do so, saving time for themselves and those responsible for that area. There is thus an optimization of the communication process within the company, aided by the ability to access any information at any time. This is the democratization of data, which increases efficiency within the company, saving time lost to non-value-added activities such as preparing data reports and subsequent presentation.

1.4.4. Real-Time Decision-Making

In data lakes, the data do not require filtering, cleaning and processing before being stored. In a normal DW, the data would have undergone the long and rigid process of ETL before finally being available for analysis, thus impeding immediate analysis of the collected data. With the data lake, the data are immediately usable and ready for analysis also aimed at extracting information for real-time decision making.

Moreover, in the data lake, being schema-on-read, data are not stored according to a rigid and fixed structure as in the data warehouse. All this makes analysis much more flexible and complete since data of any type, form, and from any source can be used for analysis. You do not put a limit on the data accepted to be stored, so you also do not put a limit on the information that can be extracted from lakes. In terms of query languages, the lake is also more suitable for use in Industry 4.0. In fact, the data lake

limits the use of SQL, but extends to a variety of languages to allow user interaction with the data.

When it comes to a rapidly evolving corporate environment, like Industry 4.0, the ability to set goals in real-time based on the data gathered is crucial. The management can instantly examine any information related the supply chain, demand, operations, sensors, IoT, and many other topics by always having industrial data on hand. Additionally, it is adaptable enough to swiftly recognize and address pressing problems and difficulties as well as respond to modifications in technology, client demands, and supply chain operations.

1.4.5. Inexpensive To Collect

The Extraction, Transaction and Loading process of traditional DWs requires filtering, cleaning, duplicate elimination and categorization of data into clusters. These are all activities that are not necessary for the data lake. The savings therefore in time and cost grow as the amount of data to deal with increases; in a context such as Industry 4.0, where the amount of data is huge by definition, the data lake again comes across as more efficient and effective than the DW.

Data lakes give us two advantages: cost containment and analysis performed with more complete information leading to increased revenue. From a study by the University of Aberdeen (2019), it is possible to quantify the monetary benefits of implementing data lakes. Revenues have increased by 9 per cent as a result of replacing the data warehouses, benefiting from the introduction of a data lake.

Regarding the shop floor of companies, the variety of data formats is a huge problem for Industry 4.0 and data warehouses. There can be a variety of data types, it can vary from tool to tool: data can be text files, images, videos, discrete or analog signals, and data can have different storage structures. Because you receive a lot of data of different formats and with different structures from all the tools on the shop floor, ingestion of the data into a canonical structured database can be a problem. The data lake, on the

other hand, by nature allows the storage of such data without incurring into that many problems.

The method by which these large unstructured databases are managed is through metadata: data associated with the data itself that allows each element to be uniquely identified.

The advantages offered by data lakes in the various theme are listed below:

- Data are not adapted to predefined structures, but data are stored and subsequently used in their raw state.
- The full data is processed, even in parallel with map-reduce technology.
- There is the possibility to process data in real-time, as well as end-to-end analytics, also using data of different formats and structures about the same process to gain "information" about a process. Typical applications for this technology include solving problems involving associative rules (the state of a technological object depends on from related events), decision trees (understanding of the causes of marriage, downtime, as well as incidents at enterprises), genetic algorithms.

1.4.6. Conclusion

In conclusion, we can say that data lakes represent a trend within Industry 4.0. The lake contains huge amounts of data, always accessible by anyone and allowing flexible analysis of all kinds. This can help improve the company in many ways: improvement of time and cost efficiency, more control, optimization of communication processes, and so on.

1.5. Data Management and Governance in the Data Lake

In the case of using data for business-critical activities, it is critical to focus as much as possible on data governance.

We have at the antipodes two governance approaches: one very rigid and structured that limits the possibility of making mistakes, the data warehouse. And another completely unstructured, Hadoop.

On both sides we have disadvantages: regarding data warehouse, as discussed in previous chapters, we find an incompatibility with big data management. As for Hadoop, a pure "data dump" is considered too risky for the governance of data critical to business success.

In fact, uploading unmanaged or audited raw data, Hadoop, even if it remains efficient with big data, often when accuracy is a critical success factor fails to optimally ensure the value of the data.

Once again, the data lake can help companies to join the flexible management style of Hadoop and the accuracy of a DW. In the following section, some data governance approaches for the data lake will be examinated.

1.5.1. Postponing at a later time

Some companies simply postpone the challenge, putting it off until later. They upload the raw data as it comes in from the sources, and then later process it when it is time for analysis. It is often used jointly with Artificial Intelligence, which discovers trends and relationships between data present in the dataset. Obviously, there is the risk of incurring into a partial data swamp: some zones of the data lake might be constantly ignored by AI, moreover, some errors or spurious data might mislead the analysis.

1.5.2. Using products on the market

Another trend is to use DW management tools on the market. Obviously, they will have to be adapted to data entry for data lakes, but at least this avoids storing spurious data within the dataset. Unfortunately, some of the flexibility characteristic of ingestion via Hadoop is lost. Costs increase both in governance and in data uploads. In fact, the time gets longer to move data every time a query needs to be run. What's

more, some of the features that benefit the data lake in managing big data are lost with this solution.

1.5.3. Writing custom scripts

This third method consists in customizing ad-hoc codes for governance. It is a widely used option, although it remains very difficult to implement. Data scientists need to be very knowledgeable about Hadoop to do a proper job, as applications, filtering, analysis and management processes will need to be linked. This option, in the case where qualified personnel can be depended upon, represents the most economical for the initial stages; as the data lake grows, it will be necessary to add new scripts or update existing ones, thus slowing down the entire process.

1.5.4. Build a data lake management application.

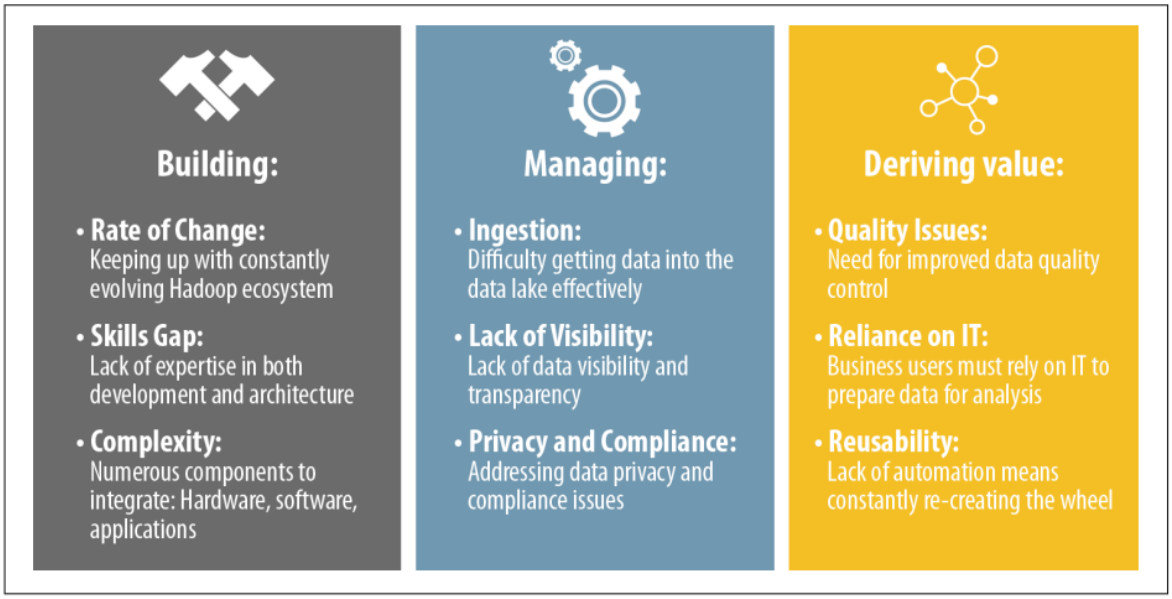
The last alternative is to build an ad-hoc platform for entering various types of data into a single data lake. This way you control the entire life of the data, from the time it enters until the time it is extracted. You have quality control of the data at the initial stages, you can catalog the data, manage extraction, and facilitate analysis.

Thus, building an ad-hoc platform allows you to take full advantage of all the benefits inherent to data lakes, while avoiding the risks discussed in the first section of this chapter.

1.6. The challenges of building, managing, and deriving values from a data lake

According to C. Giebler et al. (2019), a general strategy for the implementation and management of data lakes is absent in today's literature. A. LaPlante et al. (2016) given the presence of this lack identified what are the challenges for building, managing, and getting value out of the data collected in data lakes as shown in Figure 1.

Figure 1 Challenges in managing a data lake – O'Reilly Architecting Data lakes



1.6.1. Building

Three main obstacles must be taken into account when building a data lake:

- **Rate of change in the technology ecosystem**: the Hadoop ecosystem is dynamic and constantly evolving with several modules developed by the community. Following updates from the open-source community requires qualified personnel and time. Hadoop modules are constantly changing with new features and solutions being proposed by developers.
- **Scarcity of skilled personnel**: data lake and Hadoop are still emerging technologies that have not reached maturity yet. This requires skilled developers to design an architecture that guarantees excellent data management. These types of skills are currently difficult to find on the market and require high financial compensation. A CIO survey found that "a gap analysis highlighted that there is an increasing demand for skilled IT personnel and an offer that is not growing at the same speed. [...] 40 per cent of CIOs said they had a skill gap in information management. " (CIO, CIO Agenda Survey, 2016)

- **Technological Complexity**: The final challenge of building a data lake concerns the complexity of implementation and the technology itself. To build a data lake, it is necessary to make applications, software, and hardware interact, integrating the whole system. Hadoop itself requires integrating different modules to provide performance and utility.

1.6.2. Managing

Once the data lake is set, it has to be properly managed. In order to do it the IT department has to take care of the following aspects:

- Develop and maintain a data catalog that allows all the stakeholders to have access to data
- Update the data lake with new modules, functionalities, and data sources
- Manage and control access policies to data
- Be sure to respect regulatory policies about the use of data

All this requires periodic checks and updates of the system, which leads to the challenge of managing a data lake:

- **Ingestion**: this term refers to the process of saving data in the Hadoop file system. Performing a managed ingestion is of paramount importance, as in this phase it is also possible to perform quality checks on the data in a distributed manner before it is saved. Since not all data within a data lake are the same, flexible governance rules are required, which must be adjusted as new data sources are ingested. Collecting huge amounts of data could occur to find duplicates among them. It is therefore necessary to develop rules for the elimination of these duplicates. The higher the amount of data the higher the management difficulty will be.

- **Lack of visibility**: The data collected in the data lake must be visible to end users. The absence of transparency within the data lake can result in a problem that makes data collection vain.
- **Privacy and compliance**: when dealing with industrial data, it is of paramount importance to avoid privacy and compliance issues. Proper data governance must ensure that the company is secure and that only authorized users have access to the data avoiding exposing the company to risk.

1.6.3. Deriving value

The last challenge companies face when implementing a data lake concerns the extraction of value from the collected data. This becomes almost impossible when governance rules are not used to manage the data, as it becomes impossible to determine the quality and history of the data. In the absence of data lake management, users will have to rely on the IT department, which will overload itself with requests, slowing down the analysis process and increasing the cost of implementing the data lake. To solve this governance rules and automation (such as in the attribution of metadata tags) will both increase the overall quality of data and reduce the management costs.

1.7. Hadoop

Today we have several sources from which data can come, such as I4.0 processes, Internet IoT sensors, social media posts, reviews, and streaming data from the web. Coupled with this, in recent years is increasing awareness about the usefulness and the value of data when exploited to improve business decisions. As a consequence, the amount of data that is collected is increasing considerably in volume over the years. Many organizations can no longer manage data flows with traditional Enterprise Data Warehouses (EDW) because data analytics and strategic business intelligence needs

cannot be fully met. In order to solve these problems, many organizations are moving towards Apache Hadoop.

According to C. Lam et al. (2010) definition:

"Hadoop is an open-source framework implementing the MapReduce algorithm behind Google's approach to querying the distributed data sets that constitute the internet."

Since it is expected that a large amount of data is stored in a data lake, these cannot be processed altogether because it would take too much time and computing power with traditional methods. So, "the MapReduce algorithm breaks up both the query and the data set into constituent parts. The mapped components of the query can be processed simultaneously -or reduced - to rapidly return results", improving the analysis of data compared to traditional EDW.

Hadoop ensures the exchange of data between different applications and represents the destination of the collected data by the corporation.

C. Lam identified four main advantages of using Hadoop compared to traditional data warehouses.

Hadoop is:

- Accessible: Hadoop can be programmed to run on a wide range of machines or cloud computing services
- Robust: it is designed to work on basic hardware with the hypothesis of frequent machines malfunctioning. This provides Hadoop with high failure and malfunctioning resistance. This technical feasibility joined with its open-source nature of it lead to the diffusion of Hadoop to meet big data challenges
- Scalable: Hadoop is an easily scalable framework with a linear trend for adding computing power or data capacity
- Simple: Hadoop enables users to rapidly write effective parallel code.

According to our research, we can add two further advantages to the one stated by C. Lam:

- Cost effective: according to A. LaPlante "Hadoop can be 10 to 100 times less expensive to deploy than traditional data warehouse technologies" when dealing with enormous data volumes.
- Extensibility: because of the open-source nature of the project, over the years a large community of developers has developed tools and modules that can offer many additional functions to Hadoop. These can be management, governance tools, or others that help in the handling of data from the ingestion up to the discovery.

Table 2 resumes the identified advantages of using Hadoop architecture.

The data lake concept is closely connected with Apache Hadoop and its open-source project ecosystem. One of the main advantages enabled by Hadoop is to allow the loading of structured or unstructured data without making any modifications to the data before the loading itself. Hadoop Frameworks have to be created ad-hoc based on the context, with little preparatory work required. This leads companies to delay data cleaning and schema development as much as possible, making this system cost-effective compared to traditional EDW. This significantly reduces also the time needed for the system set up and only later, when a business need emerges, will be carried out data cleaning or conversion operations.

Table 2 Hadoop advantages compared to traditional data warehouse

Accessibility	Robustness	Scalability	Simplicity	Affordability	Extensibility
Open-source program compatible with many systems	Low minimum requirements and resilience to malfunctions	Easily scalable framework in storage and computing power	Easily coding with parallel code	Open-source program with no licence needed	Easy integration of additional modules

1.7.1. Hadoop tools & modules

Additional modules developed by the community are numerous and under constant update. In the Hadoop community, special mention should be made to the Apache Software Foundation, a community of thousands of developers that successfully collaborate to develop freely available enterprise-grade software. Here are some examples of freely available tools that can be used with Hadoop.

1.7.1.1. Apache Sqoop

Apache Sqoop is a tool designed for efficiently transferring data between Apache Hadoop and structured data stores such as relational databases.

1.7.1.2. Apache Storm

Apache Storm is a free and open-source distributed real-time computation system. It is a system for processing streaming data in real time. It adds reliable real-time data processing capabilities to Hadoop.

1.7.1.3. Apache Hive tables

Apache Hive tables is an open-source data warehouse system for querying and analysing large datasets stored in Hadoop files.

1.7.1.4. Apache Atlas

Atlas is a scalable and extensible set of core governance services. Apache Atlas provides organizations with open metadata governance and management capabilities to catalog their data assets and classify and govern these assets.

1.8. The importance of metadata

One of the main utilities of data lakes lies in loading as much data as possible in the first step and then going on to search for all the data needed for the analysis. So, making the data easily available is of paramount importance. This is why metadata is fundamental and is used within a data lake. But what is metadata?

According to J. Riley (2017)

"Metadata, the information we create, store, and share to describe things, allows us to interact with these things to obtain the knowledge we need. The classic definition is literal, based on the etymology of the word itself: metadata is data about data".

Metadata is therefore essential to the proper functioning of a data repository, enabling the correct sharing, classification, and identification of data within the database. It is therefore important in the data lake building phase to adopt a data management strategy to eliminate the costly data preparation phase typical of EDWs.

This process allows the smooth loading of data into the data lake, associating with each of them numerous metadata tags that provide information about their nature, format, and content. If correctly executed and applied to the context of use, this allows an easier and simpler identification of data within the repository.

As the number of sources that are used to extract data increases, the importance of the metadata associated with them increases too, as they will have different schemas, nature, and contexts of use from each other. This is important in EDW management as well as in data lakes due to their nature.

This metadata will then be used to "create order" within the metadata catalogue, a tool where metadata is stored to allow easy access by data analysts. As soon as a piece of data is saved within the repository, the corresponding metadata must be saved on the data catalog, in order to aggregate data that share the same characteristics, such as format, the scope of use, relationships, or provenance.

However, the number of metadata that can be collected and associated with a piece of data is almost unlimited. The selection of the right metadata model and data catalog will be crucial.

The process of annotating the data with the various data tags is carried out in the ingestion phase. This annotation can be done manually or automatically, using specific tools. Since large amounts of data are extracted and stored in data lake environments, automatic metadata extraction is a topic of great interest in the literature (C. Giebler, 2019). The main lack of data lake frameworks, such as Hadoop, is the absence of features for management, governance, and quality checks for metadata, as well as a metadata layer. Numerous metadata models or supplementary tools (most are developed by the Apache Foundation, such as Apache Atlas) have therefore been developed for this purpose, which not only makes it possible to extract data from source, but also to automatically annotate metadata on the basis of the intrinsic content of the data. By using these tools, the initial set-up process of the data lake is greatly simplified, with a simplification of the process of adding new data sources and with a reduction of the setting-up cost. In this way, it is possible to save all data within the data lake and only then, in the search phase, it will be possible to find the aggregated data set according to one or more metadata.

According to C. Quix et al. (2016), one of the most critical issues to be taken into account in the management of metadata is the use of semantics, as most queries rely on keywords. Some metadata tags may be used to indicate several different attributes or conversely different tags may be used to indicate the same thing. This only creates redundancy and confusion within the data lake, further emphasising the importance of proper data lake governance. As soon as the data is ingested, the corresponding metadata must be annotated because data without metadata will never be found and used. This allows data users to be aware of the data in the repository, and to more easily find and aggregate data of interest when querying the data lake.

1.9. Metadata classification

Considering the importance of metadata, it is also important to define the type of metadata that can be found within a data lake and how these are organized.

After an analysis of the literature, we identified two widely cited and used metadata classifications, especially for the application of metadata models in data lakes. The first one is the one proposed by A. Oram (2015) which is focused on the functional aspect of metadata. The second one is the one proposed by Sawadogo et al. (2019) which classifies data based on the structural metadata types.

1.9.1. Functional metadata

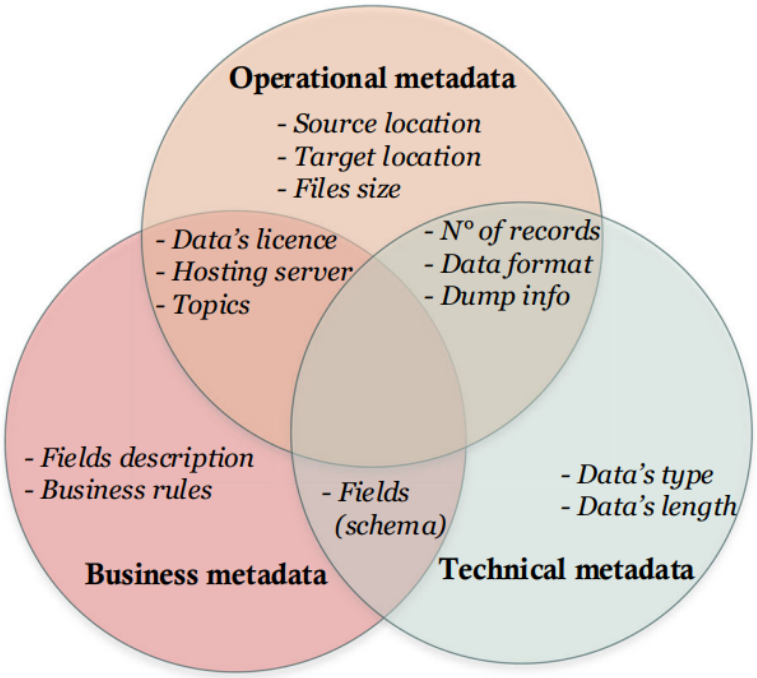
A. Oram's (2015) classification distinguishes different types of metadata according to the way they are collected. We distinguish between:

- **Business metadata**: in this category we find all those semantic annotations that make the data more comprehensible, using business terms. This metadata explains to the end user what the data means, to make it easier to find within the repository and more understandable. This metadata includes names, business terms, integrity constraints, descriptions and so on. Business metadata are usually defined a priori by users, during the ingestion phase.

- **Operational metadata**: this metadata group indicates information that can be automatically extracted during the ingestion phase. This provides information such as the size of the file, its location in the system, the source, and the number of records. This category also includes metadata describing process information such as the number of errors encountered in the extraction or the incorrect ingestion of certain data.
- **Technical metadata**: they describe the format (such as raw text, a JPEG image, a JSON document, etc.), structure, or schema of the ingested data. The features of the data structure include names, kinds, and lengths. Therefore, this metadata represents the structure and form of the dataset.

As shown in Figure 2 Diamantini et al. (2018) points out that two different of these categories of metadata can overlap with each other. In conclusion, this classification, although easy and intuitive to understand, may be limiting and not very explanatory of all the metadata that may be found within a data lake.

Figure 2. Diamantini et al. (2018) Functional metadata classification



1.9.2. Structural metadata

Sawadogo et al. (2019) instead have proposed a metadata categorization by considering an extended "metadata typology that categorizes metadata into intra-object, inter-object and global metadata with new types of inter-object (relationships) and global (index, event logs) metadata". In this context, an "object" can represent any file, document, or rational table, stored in the data lake, whether structured or unstructured. So, an object can be thought as an extension of the dataset concept. According to this classification we distinguish between:

- **Intra-object metadata**: this is the metadata associated with characteristics related to a single object within the repository. It is subdivided into:
  - **Properties**: provide a general description of an object. Provide details such as object title, file name and size, date of last modification, location within the file system, etc.
  - **Summaries and previews**: give a general explanation of an object's structure or content. They can appear as a word cloud for text data or as a data schema for structured or semi-structured data. Within this categorisation, we also find **versions and representations of metadata**. When a structured data is updated, or if raw unstructured data is refined, the resulting versions of these files may be considered as "version metadata" of the source files (since they facilitate the understanding of it).
  - **Semantic metadata**: provide a textual description that makes it easier to understand the content of the data. These can be automatically associated with files based on their content or provenance. Semantic metadata is often used to find links and relationships between data with matching tags.

- **Inter-object metadata**: represent the relationships existing between the different data in the system. These links between different objects can be between 2 or more elements. We distinguish:
  - **Object groupings**: allows objects to be organised in groups. These can be generated automatically on the basis of certain intra-object metadata (such as semantic metadata or properties). An object can be part of several groups simultaneously to facilitate data association.
  - **Similarity links**: provide a measure of the similarity between two objects. Unlike object grouping, this metadata reflects the similarity between two objects based on their intrinsic characteristics. These features may be the similarity of the structure of unstructured data, or even the content of documents, such as similar records or equivalent lines of text. To extract this metadata, it will therefore be necessary to use tools that understand and compare the structure of the data and/or its content.
  - **Parenthood links**: this category of metadata indicates the relationships between objects that have been generated by the transformation of other data. These transformations are mainly carried out on structured data once the analysis is required. This is especially important if we want to use more data from a "parent" object and therefore need to go back to the source data by knowing the data lineage.
- **Global metadata**: Unlike the previous categories, global metadata are data structures intended to give a contextual layer to the entire data lake. These are therefore not information attributable to individual objects but to the entire data lake, to facilitate data searching and analysis. Here we identify:
  - **Semantic resources**: indicate knowledge bases that are used to facilitate search and analysis within the data lake. Semantic resources are e.g. ontologies, taxonomies, thesauri, etc. which allow once that a metadata

has been associated with an object, to associate also tags with comparable semantic descriptions to it. These can draw knowledge from web sources or be developed in the design phase. These metadata can also reinforce inter-object metadata by facilitating data clustering.

- **Indexes**: are data structures that facilitate data retrieval. This categorisation is similar but opposite with respect to semantic resources. These allow the user to query the data lake with word-based queries and find metadata with similar meanings. This greatly facilitates the search for data and can also be used to search for images, videos or sounds.
- **Logs**: these metadata make it possible to record data access by different users. In this way, it is possible to trace the history of accesses and updates by associating those accountable. This information can also be used to see which data has been used most frequently or updated most frequently.

This classification compared to the previous one is much more detailed and specific, especially in understanding what requirements the metadata of a data lake must fulfil.

1.10. Metadata models

Metadata can be saved in different places within the information system: in the directories of the file system, in a specific file, or in the file name. The absence of standardisation increases the difficulties for data scientists who want to query the system. This leads to the need for a standardized system for managing data and its metadata. Although it has been repeated many times in this paper an effective data management is fundamental to administrate a data lake, the literature lacks a general approach to handling them. This consequently implies the absence of even a comprehensive strategy for the control of data lakes in IoT and I4.0 environments. Some approaches for the management of EDWs can also be implemented for the

management of data lakes but are still limited by the absence of unstructured data and the need to define the data schema a priori.

The high volume of structured and unstructured data leads to high heterogeneity in data structure and metadata semantics. To solve this data management complexity and to ensure the cost and utilisation efficiency of data lakes, numerous developers have suggested metadata model frameworks for data mapping. The purpose of these frameworks is to offer a standardised and more flexible approach to data management, capturing the data semantics. According to Miloslavskaya and Tolstoy (2016), the absence of a comprehensive data management system that enables the description of the data, can turn the data lake into a "data swamp" or create a data silos structure. According to C. Giebler (2019) "Metadata management is crucial for data reasoning, query processing, and data quality management", for this reason, is crucial to extract as much metadata as possible during the ingestion phase. There are several approaches for effective data management but few of them provide sufficient detail for their application and reusability.

Also, P. Sawadogo (2020) underlined that "data has to be extracted from the data sources, it has be to clean, transformed, and mapped to a target system, and finally, it has to be loaded into a data management system where it can be integrated with other data". This process described by Sawadogo is also known as ETL (Extract-Transform-Load) process, underling the need for a comprehensive system that takes care of data from the ingestion up to the analysis.

According to our research in the literature are present two main approaches for metadata management within data lakes: data vault and graph-based method. Other methods that are used for EDW management such as head-version tables, lambda architecture, or 3rd normal form approaches can be used also within data lakes, but these cannot ingest semi-structured or unstructured data and would therefore lead to a limited functioning of its capabilities.

1.10.1. Data vault

The concept of the data vault was first applied in data lakes by Nogueira et al. (2018). It is a comprehensive system for managing data within the repository that offers good flexibility and enables also new data schemas to be simply modelled. The major limitation of this framework is that it was originally designed for data warehouses, and thus for structured data only. K. Cernjeka et al. (2018) proposed new approaches to integrate semi-structured data into data vault, but no public method exists to integrate unstructured data.

According to F. A. Eshetu (2014) data vault modelling requires three main elements:

- Hub: that is intended as a basic entity that represents a business concept (e.g. product, customer, process, machine…)
- Link: indicating a relationship or an association between different hubs (at least two).
- Satellite: which contains the descriptive information of the respective hub or link

Each satellite is associated with a unique hub or link. Conversely, several satellites can be attached to links and hubs.

1.10.2. Graph-based data models

Graph-based models adopt a graph view of the entire data lake and are based on the P. Houle (2017) data droplets model. According to this framework every object within the data lake, such as documents, tables, photos and so on must be modelled as an RDF (Resource Description Framework) graph. Each of these graphs is then combined with the others on the basis of shared characteristics (contained in the metadata) and relationships to form an overall graph of the entire repository.

Most of the metamodels used to manage data lakes rely on graph-based frameworks since it allows flexible management of the lake and allows the ingestion of

unstructured data. So, this method allows metadata schema evolution, increasing the flexibility of the data lake. According to P. Sawadogo et al. (2020), the main disadvantage of graph models is that they require specific storage systems, such as RDF or graph DBMSs, but they increase the information present in the system simplifying analysis and the research of data.

Here we will quickly see some examples of graph-based metamodels that we will be further explored in subsequent chapters.

1.10.2.1.Graph-based data models: some examples

In the literature, there are several generic graph-based metamodels that can be implemented. The main differences lie in the functionality offered, the data representation model and their complexity.

The C. Quix et al. (2016) model named GEEMS, for instance, represents the objects of a data lake such as data files and data entities. A data file is defined as a generic data source consisting of several data entities, and elements that belong to the data file. Different metadata may then be attached to each of these entities. J. M. Hellerstein (2017) instead proposed GROUND, a metamodel developed considering the sources from which the data may originate with the basic ABCs (Applications, Behaviour, Change). This model makes it possible to identify three levels of metadata: metadata properties, data usage history and data versioning making it evident that many metadata categories previously explained are not considered. A final model we report is the one developed by R. Eichler et al. (2021) HANDLE (Handling metAdata maNagement in Data LakEs) where each data entity is associated with tags representing zones, granularity levels or classifications. This enables the association of different tags providing a comprehensive metadata management for data lakes.

1.11. Metadata models requirements

Without an adequate and effective metadata management system, as explained, a data lake risks turning into a data swamp. How, then, does an organisation understand which metadata model is most appropriate for it? According to our research, there is no objective way of evaluating different metadata models to see which one is more suitable in one context or another. The risk is that by choosing the wrong metadata model, data could enter into the data lake and then it will not be used, or the analyses performed will not take into account all the data of interest.

1.11.1. Sawadogo

Given this shortcoming, Sawadogo et al. (2019) proposed a method for comparing different metadata models on the basis of the features that a data lake must have in order to be considered comprehensive: Semantic enrichment, Data indexing, Link generation and conservation, Data polymorphism, Data versioning, and Usage tracking as resumed in table 3.

Table 3 Data Lake needed features

Semantic Enrichment	Data indexing	Link generation	Data polymorphism	Data versioning	Usage tracking
Textual information to make the data more comprehensible	Allows the use of keywords or patterns to search data	Generates links correlating data that are related	Allows multiple representations of a single data to be saved	Handle update operations preserving previous states	Recording of users iterations with the data lake

- **Semantic Enrichment**: this function allows to add textual information to the data, such as the title, descriptions, and tags, to make the data more comprehensible to the user. Knowledge bases are usually used for this function (such as ontologies). Quix et al. (2016) stated that semantic metadata could be the basis of link generation between data.
- **Data indexing**: it allows keywords or patterns to be used to aggregate data and create a data structure based on specific characteristics. This function thus makes it easier to search for data within the data lake by associating words similar to those the user is looking for with those that describe a specific data.
- **Link generation and conservation**: this function consists of generating links and correlating data that are related or have similar characteristics. This can be done manually by correlating data sources during the design phase or automatically on the basis of the intrinsic characteristics of the data. These links will then be shown to users querying the data lake, increasing the spectrum of analysis or showing clusters of data in an automatic way.
- **Data polymorphism**: if a data is transformed to be adapted to a new context, there must be a reverse function that allows to go back to original state. This function allows multiple representations of a single data to be saved. In this way the multiple representations of the same data are allowed at different levels of detail or structure. It refers to the fact that I can find the same data more or less structured, altered or adapted depending on the context. This is useful when editing unstructured data to understand the origin of the resulting structured data or to identify and correct possible errors. You must understand how much the data has been altered to fit the structure of the context in which it belongs and must be able to get back to the raw state as well as transform it following the needs of usage.

- **Data versioning**: this functionality expresses the system's ability to handle update operations and support changes while preserving previous states. This functionality relates automatically two or more data, in which one is the latest updated or modified version of the other. This function is especially important when there are errors in the dataset in order to retrieve data versions that do not have inconsistencies.
- **Usage tracking**: allows the recording of iterations (creation, access, and update of data) between users and the data lake. This function increases the transparency of the data lake and makes it possible to trace those responsible for modifying or updating files. Usage tracking is therefore essential when managing sensitive data or complying with privacy policies.

The usefulness of each of these functions depends on the context of use, so the selection of the right metadata model will depend also on this. For some organisations, one feature may be essential while another may be unnecessary.

The more features a data lake has, the more it can be considered generic and thus applied in more contexts. On the other hand, a metadata model with a limited number of features can be successfully applied in specific contexts.

1.11.2. Eichler

Another set of features was proposed by R. Eichler et al. (2020). These are feature of a data lake used to enlarge the analysis and understand how a data lake needs to be shaped to be integrated into Industry 4.0 environment. With the feature-based approach, the model is created to accommodate a predetermined set of features. Features are derived from use cases for metadata management. The metadata model would be complete if it supports every feature on the list, which includes all important features for metadata management according to researchers. We test the limitations of the metadata models using the four constraints imposed by the scenario that is being given. The metadata models must enable data lake zones and be flexible in the creation

of metadata characteristics for metadata objects to reflect the widest range of information for this use case. Table 4 resumes the identified features proposed by R. Eichler et al.

The first requirement is adding **metadata properties** to ensure flexibility, so modelling the metadata as flexible as possible. According to R. Eichler et al. (2020)

- Metadata can be stored as metadata objects, properties, and relationships
- The number of metadata objects per use case is unlimited
- Each metadata object can have an arbitrary number of properties
- Metadata objects can exist with or without a corresponding data element
- Metadata objects and data elements can be connected
- Data elements can be connected

The second criterion is the **granularity levels**, the capacity to gather metadata at various granular levels, maintaining flexibility in terms of the level of detail and distribution of metadata. The approach facilitates the inheritance of metadata at granular levels: technical metadata, for instance, that is added at the schema level also applies to more specific data items like tables, columns, rows, and fields. The metadata model was created to accommodate data lake characteristics because it was designed for metadata management in unstructured contests.

The third functionality is **the data zones**: the model must accommodate the idea of data lake zones because the majority of metadata is gathered on particular data objects that are categorized into zones. This means that metadata should be distinct between zones, giving metadata allocation more freedom.

Finally, it should be adaptable meaning that it can incorporate any **categorization** in the form of labels, such as MEDAL's intra, inter and global labels. This makes it easier

to quickly determine the context of the data. Additionally, it may be used to verify that all kinds of metadata are being gathered.

According to the author, these four needs make up the new set of general specifications for a generic metadata model in the context of data lakes.

The granularity entity makes it possible to collect metadata at various granularities. These levels are strongly related to some sort of data structure. Examples of granularity levels include object, key, value, or key-value pair instances in a JSON document. But the level of detail is not just for structured data. Videos, for example, fall under the category of "unstructured data," although one might want to capture metadata on specific video frames. There would be a video level and a frame level in this scenario. When choosing granularity levels, domain knowledge can be useful because it's frequently important to know, for example, whether the metadata pertains to the content of a single frame or an entire film.

The zone entity is a label on the data entity that provides details about the position of the data element in the zone architecture of the data lake. The degree of the data's alteration is immediately visible through it, depending on the zone specification. As enumerations for the zone, the various zones are modelled. The zone enumerations and their associations must be modified to use another type of architecture. Every data element must have exactly one zone indicator according to the model since all data recorded in the other zones will have a corresponding data element in the raw zone. The indicator also shows the fact that the same entities present in different zones are linked to the matching data element in the raw zone by a link entity. The link stores the information about the source from which the data was imported into the zone as well as the appropriate timestamp. The name of the original source or a zone may be included in the importedFrom attribute. The connection and importedFrom attributes make it possible to follow the passage of the data through the zones.

The label for the categorization entity is chosen in accordance with the context of the metadata element. A metadata element storing any type of access information will have an operational label since, for example, access information is core metadata and operational metadata as described by.

Table 4. Eichler 's features of a Data Lake

Metadata Properties (flexibility)	Granularity	Data Zones Identification	Categorization
Modeling the metadata as flexible as possible.	The capacity to gather metadata at various granular levels	Metadata should be distinct between zones, giving metadata allocation more freedom.	A metadata element storing any type of access information will have an operational label.

1.12. Data catalog

According to Gartner (2017)

"a data catalog maintains an inventory of data assets through the discovery, description and organization of datasets. The catalog provides context to enable data analysts, data scientists, data stewards and other data consumers to find and understand a relevant dataset for the purpose of extracting business value".

Data catalogues are so the data lake tool used by end users. Thanks to this tool, it is possible to search, discover, manage and aggregate data within the data lake, despite the fact that these come from different sources. This tool shows all the data that is stored in the system, displaying all the data as if it came from a single repository, since with data lakes we have no silo structure. The data catalog provides a 360° overview

of the content of the data lake by giving information on the documents, recorded measures, semantics and respective metadata of each file contained in the system.

According to A. LaPlante et al. (2016) once the data is saved within the system, people will use this data to derive business insights. To do this and to aggregate the data correctly, it will then be necessary to have information about, quality, properties, and input history within the data lake or transformation history of individual data, profile, and metadata for all types of data, business, technical, and operational. This information must be displayed to users in a way that facilitates understanding and enhances the user experience to provide an effective and easy-to-use system. This is precisely the function of the data catalog.

The goodness of this data search system is strictly dependent on the metadata and the metadata model chosen in the design phase of the data lake, as the data catalogue needs numerous and precise metadata to function properly. The more accurate this search system is, the less time data scientists will have to spend preparing and cleaning data, making the process leaner and more agile.

The use of an effective data catalog, therefore, allows users to carry out analyses and searches semi-autonomously, without requiring the intervention of the IT department, thus reducing the time and costs for analysis. The quality of a data catalogue will therefore depend on the efficiency of data management and the clarity and usability of its user interface.

According to E. Zaidi et al. (2017) "through 2019, 80% of data lakes will not include effective metadata management capabilities, making them inefficient", emphasising the importance of an effective metadata model and data catalog to support analysis.

Different solutions from different manufacturers such as Microsoft, IBM, Oracle, etc. are present on the market, as well as open-source projects such as Amundsen or Delta Lake. The problem with commercial systems is that they are very often closed and proprietary systems, with their own glossaries and metadata languages. It may

therefore be better to develop a proprietary system that allows the use of proprietary vocabularies.

1.13. Literature gaps

Since data lakes are a relatively new concept, the literature is incomplete and vague regarding approaches for implementation in industrial contexts. The same management models we have seen do not provide detailed aspects for design and implementation, moreover, they often explain theoretical frameworks with few real application cases. This leads companies to research and evaluate alternatives to choose the best approach.

1.13.1. Data lake architecture

The heterogeneity of these approaches is a major problem for data lake architecture. This is further fueled by the fact that there are no objective ways of comparing the different proposals offered by academics. This leads to the need for analysis to understand the differences and similarities between the various approaches. The absence of a "best practice" can be a problem but also an opportunity as organizations, after careful evaluation, can select the model that best suits their context of use. In addition to this, the frameworks discussed in the previous chapters only represent conceptual approaches, without giving details or examples for the concrete implementation (such as modelling or data lake infrastructure) of the different solutions. Once the best solution has been identified, it will then be necessary to concretely define a generalized architecture.

1.13.2. Data lake governance

Another aspect to be further explored is data lake governance. Data lakes offer an inexpensive, scalable and flexible alternative to more traditional EDWs. While this is seen as an advantage, it can also be a problem at a time when data quality and accuracy

are crucial for the business. In this context, literature is lacking for an effective data governance approach, to structured data lake with the rigour of a traditional data warehouse. This will require the definition of new concepts of flexibility and access to data specifically designed for the data lake.

1.13.3. Data lake maintenance

To the best of our knowledge literature lacks about post-implementation management of data lakes. E. Scholly et al. (2021) noticed that metadata maintenance is a wide-open issue. The literature, offering mainly theoretical frameworks lacks practical information on how to handle the input of new data sources, new metadata categorisations, the possibility of changing metadata models if the one currently in use turns out to be limiting for the business, how the functioning of the data lake changes as the amount of ingested data increases, or regarding the management of obsolete data to adhere with the FAIR principle.

1.13.4. An holistic approach

These shortcomings lead to the absence of a comprehensive strategy for the implementation and utilization of data lakes. Comprehensive strategy means a strategy for building, modelling, managing, and extracting value out of them, considering the interdependencies of these aspects and providing a holistic approach. For instance, the same metadata models are seen above lack methods for an effective and integrated querying process. It is therefore necessary to understand the interdependencies between these processes in order to provide an integrated and holistic approach to the adoption of data lakes in practice.

1.13.5. Industry 4.0 & Internet of things

Today's literature lacks narratives concerning the implementation of data lakes within I4.0 and in IoT-related environments. Despite this, it is well known that huge amounts

of data are produced in this application environment. These can then be exploited to perform analyses for process and product improvement. Statistical surveys show that the number of IoT sensors capable of capturing and collecting data will increase by 1 trillion by 2030 (M. Chen, Related Technologies in Big Data, 2014), making it clear that it is necessary to intervene with the most appropriate tools to manage data.

The collection of a large amount of data and the nature of these (structured and unstructured) make data lakes a perfect tool for data collection and research in industry 4.0 environments.

At the moment, the so-called 'industrialization' of data lakes is still missing. This term refers to the provision of a software layer linked to the metadata system, which enables non-IT users to extract, group and analyze their data of interest. Nevertheless, this layer of software must not turn into another complex system, since its purpose is to allow data analysts to work without the support of the IT department.

2 Data collection with interviews - The MADE experience

2.1. Introduction

To select the most appropriate metadata model for data lake management, information about the use of data in I4.0 contexts is required. This information must concern the ingestion, the purpose of the data collection and detailed information about the utility of different metadata categories. Searches in the literature have not provided us with enough information to draw conclusions on this issue, thus necessitating further investigation in this field.

To do this, we decided to turn to MADE, the Politecnico of Milano's I4.0 competence centre, to collect data for our project work. This collaboration allowed us to interview different area managers each one specialized in a different field of I4.0. The aim is to identify the needs and requirements when implementing a data lake in I4.0 environments. These interviews, which lasted on average one hour, were divided into two main phases. The first phase was cognitive, in which questions were asked about the as-is situation in the area of competence. Here, questions were asked about the data that is collected, the analyses that are performed, how the data is stored and managed. This allowed us to gather a lot of further information on how the organisation currently works, the machinery used and the IT service providers. The second phase of interviews instead, was preceded by the presentation of the metadata classification models of Oram and Sawadogo et al. Once this was done, and after leaving room for questions to clarify the two models, we proceeded by asking respondents to rate the different categories of metadata and to comment on their usefulness. Understanding what metadata is needed in the different areas is crucial to understand which metadata model better suits in this context. The only missing information from these interviews

concerns the current management of metadata for the databases used in the different areas since those responsible had no information about it.

2.2. MADE 4.0

MADE is a digital and sustainable factory that supports manufacturing companies on the path of digital transformation to Industry 4.0. It provides a broad panorama of knowledge, methods and tools on digital technologies ranging from design to engineering, production management, delivery, and end-of-life management.

With its large 2,500-square-meter demo center, training rooms, co-working, and meeting spaces, it represents a one-of-a-kind organization. The special growth path toward digital transformation setup is based on three steps: informing and showing Industry 4.0 technologies, explaining through ad-hoc training activities, and finally transferring and implementing through projects the technological solutions.

MADE - Competence Center Industry 4.0 is a technical interlocutor to turn to manage innovation activities, technology transfer, applied for research and assistance in the implementation of Industry 4.0 technologies, maintain a highly competitive level by restructuring organizational and business models and the strategy of companies.

2.2.1. Area 1 - Virtual Design and New Product Development

"Virtual Design and New Product Development" is MADE's demonstrator where the most innovative methodologies and tools for digital new product development are presented and experienced. The objective of the demonstrator is the digitization of typical new product development activities and interactions with other business functions.

The starting point is tools that allow the user access to product data, appropriately controlled according to the user's role. Through a graphical interface that can also be accessed from the Web and mobile applications, it will be able to plan the necessary

activities, share the progress of the project with collaborators, and access information related to the product family. In this way, product know-how rests on a solid digital foundation and is strongly geared toward collaborative development.

2.2.2. Area 2 - Digital Twin and Virtual Commissioning, Logistics 4.0 and Lean Manufacturing 4.0

The area represents a true Smart Factory where digital technologies are integrated with a Lean vision of logistics and production processes. This area aims to demonstrate to the user the benefits of using digital tools such as Industrial IoT, Cloud, Data Analytics, Collaborative Robotics, Virtual Commissioning, and Product and Process Digital Twin in a real production line. The cluster presents the vanguard of 4.0 technologies to support the Enterprise System by offering different innovative views divided into 3 use cases: first, we have a Digital Twin that enables in-depth analysis of both production process and product characteristics, preventing design errors and predicting final performance. At the same time, it becomes an enabling technology for new ways of analysis and business models. The second use case is the Lean 4.0 methodical approach, which defines a lean and agile factory in which all 4.0 technologies can be implemented and fully exploited while avoiding digitization/automation waste. Finally, logistics 4.0 leverages IoT, RFID and advanced automation technologies to create efficient, coordinated material flow and information flow useful for continuous control and improvement of the System.

2.2.3. Area 3 - Collaborative Robotics and Intelligent Worker Assistance Systems.

The area has two thematic sections that show the role of humans in the digital transformation process that is shaping the industry. One thematic area is devoted to collaborative robots, which are industrial robots specifically designed to be able to operate safely in the presence of humans. Collaborative robotics allows for simplified installation of machines, requiring no physical protective infrastructure and therefore

less space occupation in manufacturing environments. The second thematic area is devoted to "smart" systems for assisting Operator 4.0. These are architectures consisting of hardware devices and the software that manages them, integrated to support humans, who perform traditional factory operations, assembly, training and retraining, all facilitated and augmented by technology. The hardware devices mentioned above are the so-called wearables, i.e., elements equipped with artificial intelligence capable of enhancing interaction with the surrounding environment through their respective dedicated software. These are what are generally known as cyber-physical systems, i.e., apparatuses capable of making enhanced interaction.

2.2.4. Area 4 - Quality 4.0, Product Traceability and Additive Manufacturing.

A digitized production chain that combines traditional processes and new production paradigms is the focus of MADE's Quality 4.0, Product Traceability and Additive Manufacturing area. From design to the final distribution network, product and information travel hand in hand in new ways. The product is increasingly complex and personalized, and the information increasingly rich and accessible.

The path within this area leads the visitor through state-of-the-art solutions and technologies articulated in three interconnected themes. The first concerns additive manufacturing, or 3D printing, as the enabling technology of the new digitized factory. Indeed, additive manufacturing enables the design and manufacture of highly customized products with completely new geometries and performance that are difficult or impossible to achieve through traditional technologies. The second theme focuses on new challenges related to quality control of increasingly complex and customized products. The solution is to move, as much as possible, information from the product to the process, through signals, images and videos that, collected in real-time, represent a true process signature. The increasingly rich information that travels along with the individual product is also the basis of the third and final theme. Here

the focus shifts from process data to product codes that enable their management and tracking throughout the supply chain through new serialization 4.0 methods.

2.2.5. Area 5 - Smart Monitoring and Control ff Industrial Processes, Smart Energy Monitoring and Control, Smart Maintenance

The technologies and tools that characterize Industry 4.0 find their application in the area in three areas of industrial plant and machinery management: monitoring and control of industrial processes, energy monitoring and control, and maintenance.

The area includes two Machine Tools and Service Plants for the distribution of electricity and compressed air; in addition, the area also includes demonstrations of remotely monitored Plants at Whirlpool plants in Italy.

2.2.6. Area 6 - Industrial Cyber Security and Big Data Analytics.

The use cases of this area are the technologies and tools that characterize Industry 4.0-such as innovative production systems, increased plant connectivity and a focus on resource use efficiency-find data. Data governs existing plants, can simulate future solutions, and create virtual environments, all thanks to a dense network of communications between the different technologies employed. The Industrial Cybersecurity and Big Data Analytics area aims to provide evidence on how data is now a strategic element in increasing business competitiveness and how it must be appropriately managed and protected as a result.

2.3. The Interviews – 1st phase

As can be inferred from the previous chapter, data is of great value. The following part will present the results of the survey conducted with MADE area managers, aimed at understanding what data they collect, how they use it, how they store it, etc... The results will be presented area by area following the pattern of questions used for the interviews.

2.3.1. Questions Pattern

2.3.1.1. General

1. What data are collected?
2. What are the processes that generate data? Are the data collected related to the processes from which they are extracted?
3. What are the applications and/or systems that generate this data
4. Is there also data that is purchased from outside your area of responsibility?
5. Is the data you generate used in other areas?

2.3.1.2. Collection

6. What standard/format is used for data ingestion?

7. What tools are used for data ingestion?

8. Does the data generation and collection procedure causes errors? How reliable are the collected data, can there be errors in the collected measurements? (missing data, errors in data transfer, integration of new sensors...)

Storage

9. What standard is used for data storage?

10. What tools are used for data storage?

11. Where is the data stored (data warehouse, data lake, database...)

12. In what formats are the data saved (structured/unstructured and formats eventually)

- In how many formats is a single piece of data saved on average (unstructured, in multiple structured formats...)?
- Is both real-time and batch data collected? Do you think either type of data is more important?
    i. In the first case, how quickly is it necessary to transfer data? In the second with what periodicity are they transferred?

2.3.1.3. Analysis

13. Are analyses of the data done?
    - If no analysis is currently done, what is the reason of it (lack of data, difficulty in doing analysis...)?
14. What is the purpose of data analysis?
15. What are the tools used for data analysis
16. Where is the data analyzed? In the cloud? On-premise? Have you ever encountered any issues in the data analysis process (difficulty finding useful data in the repository, data quality...)

2.3.2. Area 1

Area 1 tasks involve the product development process, they have two case studies: a marine engine and a refrigerator (consumer good). They have taken SW tools to improve and optimize the product development process. They have a virtual reality space where they show product designs, and a data-driven design station where they collect data on refrigerators and want to use it to design a 3D scanner as well.

- **What data is collected and what are the processes that generate it:**

In design, they use SW product lifecycle management. The database has a graphical interface that allows the user to interface without having to make queries. In addition, the DB manages the design process, data modification, and generates files. The sensors collect data only on the refrigerator, which is sensorized with thermo pairs for temperature and measures power. Each product design then generates data that is entered into the database.

- **Data exchanges with other areas:**

The data is not used by other areas because PLM on Microsoft Azure is required to access it. It may happen that there are data exchanges, in fact, via USB stick data is

passed with Area 4 regarding additive manufacturing processes. No data is then taken from outside.

- **What standards/formats for ingestion:**

Q-box attached to the refrigerator is a custom thing where they put different boards and use them to ingest data.

- **Errors:**

There are not many errors in the data gathered by sensors, one problem is that Thingwordls has a client part (structure the app) and a server part (database), sometimes the server part goes down, so there are some missing data.

- **What standards to store data:**

They use Microsoft SQL standard for storage, data are stored in the cloud on Microsoft Azure, two server instances on M Azure, one PTC and one Siemens.

- **What formats the data are saved in:**

They are almost all proprietary to the various SWs and neutral files from CAD storage systems. No unstructured data are collected.

- **Batch or Real-time:**

Data are saved in batches for the design part of the refrigerator, with PLM you work with check-in procedures where the data are saved so no one can change it, then you finish working on it and save it in the database. The refrigerator data is extracted in real-time from the sensors.

- **Analysis:**

No data analysis is done because there is no need, they don't know how to use it.

- **How they could be used:**

You could try a Machine Learning Approach where you don't use a pattern but look for patterns to facilitate future product designs.

- **Data catalog or database description document:**

There is no data catalog, but there is a DB usage guide.

- **Metadata management templates or patterns:**

Does not know if metadata management templates are used. They are two XML (siemens PTC) ad-hoc schemas, but they are moving to RDF

2.3.3. Area 2

- **What data are collected and what processes generate them:**

They just built the database, so is not much populated. The Hass Machine collects the processing parameters of the machined plate and traces the start of the order. The idea is to track each machine and for each order the history, so, what the processing steps were in the system. They mainly collect management data though, so not robot measurements. So, largely there is time and quantity data, such as cycle start and end time, and order queue length at the beginning of the process. All processes generate data: here, they keep track of processing times and quantities and order queues.

- **What applications generate the data:**

They have the sources (sensors on Hass machine) connected then through Modi's connecter, then get everything to Siemens Mindsphere which is their DB.

- **Data exchanges with other areas:**

They only use data from their area and do not give data to other areas, even if they would like to.

- **Errors**:

No errors have come up yet since the collection process is not running that much.

- **Where they are saved and how:**

All data are saved in the cloud. All data are structured and presented with a predefined structure proper of their database.

- **Batch or real-time:**

Data are saved afterward, certainly not needed in real-time. AGV has data in real-time but does not save it afterward.

- **Analysis:**

The purpose of the analysis is performance evaluation and possible revision. The analysis is made in batch.

- **Analysis tool:**

Aizoone made the data analysis tool, the interface is proprietary. All analysis is done in the cloud.

2.3.4. Area 3

- **What data are collected and what are the processes that generate them:**

There is no real idea of data management. They gather wearables data from sensors that measure electrical activity in the exoskeleton, which then transmit to an internal pc. Other experiments are monitoring operators efforts with and without exoskeleton, that do not generate data. Operational Robotics sends data to MADE's network.

- **Data exchanges with other areas:**

Area 3 use only local data, data are not used in other areas.

- **Where data are saved and how:**

Standards for ingestion: for exoskeleton, there is the application of sensors vendor, which collects data. Accelerometers are from x-sense. Other sensors are Delsyis. No instruments are used for ingestion. Sensors' data are saved in a local database. Instead, robotics data are saved in CIA data lake. So they have two different places of storage and struggle to perform cross-analysis.

- **What format are data saved with:**

Acceleration and electrical activity of exoskeleton are real-time data, which are local facts saved in local in a standardized way. Robotics data are ingested raw into CIA data lake.

- **Analysis:**

Analyses are done on the exoskeleton of data collected from sensors. Operator effort is measured. They aim to highlight movement effort comparison with or without the exoskeleton, in order to understand the effectiveness of the exoskeleton. There is no procedure for using the data. They aim to be able to use the data gathered for training operators to perform the right movements in their operations with the exoskeleton.

2.3.5. Area 4

- **What data are collected and what are the processes that generate them:**

Process data are collected from the two metals and polymer printers, mainly metal printers. The data collected are as follows: environmental data, temperature, laser power. They do not have built-in sensors for the moment, but they could have photographic data such as videos and photos, regarding tomograph values, so, porosity and pore size with geometric reconstructions. The only images captured are those taken by the Siemens camera on the Rockwell line checking parts as they pass through. CAD design data and process files are also saved from the printer. Each process listed above generates data, which are linked with systems that aggregate process data within an application to keep track of the process from which they come: Cefirel for methodology data, while Engine soft makes process data available.

- **What applications generate the data:**

Pheonix tomograph, Prima Industria printer, Rockwell line feeding IBM's AI system from Siemens camera photos.

- **Data exchanges:**

The design of additive manufacturing parts is related to area 1, so they exchange data with them. Data regarding manufactured parts and then post-processing data could potentially be exchanged with area 5 exchange.

- **What formats are the data saved in:**

OPCUA is the protocol used for data extraction. There is no standard for data storage, they save the raw data: the data regarding tomography has its own raw format. The same for the images, they are raw images and then the IBM server gets them from the DB where they are standardized according to their formats, so it is the partners who make the software platforms that choose the format. The outputs of the analyses, on the other hand, are tables that can be saved in multiple formats.

- **What tools are used for ingestion and where the data are saved:**

They are not yet able to extract all the data, for example, they are trying to extract data from the Rockwell machine again through OPCUA and then make it usable through Engine Soft, but at the moment they are not doing that yet. Printer and tomograph data are saved raw on local computers, as previously explained. Instead, the images collected by the Siemens camera are sent raw to the IBM DB to feed the AI.

- **Errors:**

As far as errors and missing data are concerned, there are not many, but it may happen that there are communication problems on the Siemens camera server that cause some data to be lost.

- **Real-time and batch:**

The tomograph and printer save their data in batch. Instead, the images are sent to the IBM DB in real-time. Future data collected by the Rockwell line will be in batch.

- **Analysis:**

The purpose is to show what can be done with the analysis of images for error identifications and to show that important information can be extracted from 3D

printers and also from tomographs. The goal of the area is to do analysis: analysis is done on tomography measurements to analyze the profiles of products and reconstruct their geometries. There are two ways: either directly with the proprietary software or the data are exported and analyzed with Mathlab. The part images collected by the Siemens camera are processed by IBM for defect analysis. Process data from the Rockwell machine will be passed to Engine Soft to analyze print errors, qualifying the product and process.

2.3.6. Area 5

Here they perform both predictive and condition-based maintenance. They have 3 CNC implants, one of them is a utility machine that runs the other 2. One machine is old, which not being digital native has been retrofitted, three accelerometers have been put inside it and new numerical control capabilities for predictive maintenance. The other machine is newer (digital native), in fact, there is already numerical control with sensors, and the focus is energy monitoring. Machine utilities adapt energy consumption downstream of machinery demand.

- **What data is collected and what are the processes that generate it:**

The data collected are the acceleration data that comes from the 3 accelerometers placed on the old machine, we then have the energy consumption data generated by the new machine and by the utility machine. Then there are data that they do not collect in the area, but share with the partners for the programming part: these data are the bit feed rate and the spindle speed. These data are not used for the moment, in perspective, they will be used for enriching predictive maintenance. The processes that generate data following machining are roughing and coarse-grained milling, from which acceleration and energy consumption data are collected.

- **What applications generate the data:**

Bosch accelerometers and numerical control sensors for energy consumption generate the collected data.

- **Data exchanges:**

They only use data produced in area 5 and do not take data from outside although it would be useful, e.g., from area 4.

- **What standards/formats for ingestion:**

The main standard for ingestion is the OPC-UA protocol. The OPC-UA protocol is modified by Siemens, so they have a few more functions. Canonical OPC-UA is also used for data extraction from the accelerometers placed on the old machine. For extraction of the other data, modified OPC-UA is used.

- **Errors:**

There are some technical problems with data collection at the installation level. The accelerometers occasionally vibrate a little too much and thus give problems. But the main problem is that the collection procedure is very jagged, in fact, as will be seen in item number 6, there are many partners involved in data collection, storage and presentation.

- **Where the data are stored:**

The sensors send the data to 3 gateways through OPCUA: Bosch, Siemens, and Alleantia, which then transfer it to two architectures alternately. The first option is the Mindsphere cloud on which the data is processed by doing dashboarding with the collaboration of SAP, Siemens, and Bosch. The second one is transferring data to an edge architecture where gateway data can be collected. The use of the two is alternated; however, the cloud one has too low acquisition frequency. The edge one, on the other hand, has to be set up for acquisition, but it remains very convenient because the data is also saved locally. The data is saved in a structured way, the standard goes to the discretion of the vendor in charge. As mentioned in item 5, it remains very difficult to put the data together since so many partners are involved, so

the standard difference is one barrier to the crossed utilization of the data. Another barrier is all the different places where data are located.

- **Real-time or Batch:**

Gateway data (those saved in the edge architecture) are saved in real-time since it is important for the purpose of predictive maintenance as it is used to train the algorithms. Cloud data, on the other hand, is saved in batch.

- **Analysis:**

The purposes of analysis are condition-based maintenance (novelty detection, state detection, and anomaly detection) and predictive maintenance. For condition-based maintenance, energy consumption and acceleration analyses consist of simple statistics to understand the average behavior for identifying any anomalies in the acquired values. They do linear regression trending, using normal and standard deviations with multiple sigmas and assessing the future behavior of the machines. As for predictive maintenance, they use Artificial Intelligence, but to date, it is at basic levels. The goal is to enrich the database that AI works with data regarding the bit feed rate and spindle speed.

- **The language for analysis is relative to the partners:**

SAP uses pyton. The other partners have their own proprietary tools.

2.3.7. Area 6

- **What data are collected and what are the processes that generate them:**

Data are collected in real-time from a production line that simulates the production of a brake. There are 4 stations: one to check the tightness of a cylinder, then one where a saucer is placed on the cylinder, then a machine that checks the quality of the work, and finally one where the saucer is removed. The finished products are then collected in batches of 2 or 3 pieces. We have product data and process data: the product data are serial data, i.e., information about the quality of the cylinder and saucer, which

allows us to understand whether there were problems or not. Process data, on the other hand, are the processing speed of the conveyor belts, errors/blocks in the process, and records of any human intervention.

There is also a digital tween(s) with flexing that simulates the same behavior described above and generates the same data. Here there is the processing of multiple parts and subsequent assembly of them. Some data, therefore, are collected with the physical system, while others are only simulated. The data refer to a real factory present in Italy. Tweens are from Brazilian factories and 2 other countries. The different lines are homogeneous.

There is a system of Virgil Dicomau which is a workstation with sensors, to do training and to evaluate the quality of operations, the data collected are structured.

Then there is a cyber security part where we have traffic data of the various components. The goal is to demonstrate how good network segmentation avoids cyber security problems (segmented, divided into logical areas). One problem, for example, is a crash of the production engine given by an opening of an email containing malware. Here a firewall is used to segment the various networks. In the various demos of possible attacks, traffic data is collected, it is semi-structured, classic log data (date, time...). However, they are not currently stored.

- **What applications generate this data:**

The Flexim simulator and the actual implant.

- **Data exchanges:**

There are no data exchanges, although it is believed that it would be very useful.

- **What tools are used for data ingestion:**

The digital tween data are CSV. The connection of Brembo and Rockwell plants, on the other hand, is through an OPC-UA protocol. Brembo and SAP, on the other hand, send data with Cloud Connect, a proprietary Siemens solution.

- **Errors:**

There are few errors in digital tween, in Brembo on the other hand there are numerous errors due to instrumentation.

- **Standard is used for data storage:**

Simple CSV, as they are collected, they are stored. There is only structured data.

- **What tools are used for data storage:**

Data collected in real-time are sent to a database called Rockwell's Hystorian, which is specialized for collecting temporal data, which is also useful because being real-time data it is necessary that it is not in the cloud to have the data close. Other data are sent to SAP's data lake in the cloud but remain available within the area.

- **Analysis:**

The analysis is done (prediction in case a plant stops), and graph prediction to show data and productivity indicators. For cyber security, however, nothing is done. The following tools are used to perform the analyses: SAS and SAP analytics are used for analytics, and Brembo's real-time line data are ignored. The Rockwell system has no real analytics. They have never experienced problems in research given because they never did it. They don't use data lake properly, they use it as a database and they fill it with what they need, not with everything they can.

2.3.8. Conclusions

In general, from the information gathered, it appears that the data could be used better and more especially in the analysis phase. There is a collective feeling among the managers of the various areas: everyone would like to be able to rely on larger and more extensive data sets. Acquired data often remain unused or even unsaved in various areas. With the implementation of a data lake, one would have a database shared with all areas: indeed, it has been stated by the various managers that they are interested in being able to rely on data from other areas for analysis and improvement of their work. A centralized data lake, ready to collect any format or type of data,

would also reduce the complexity and fragmentation of ingestion and storage procedures present in the MADE.

The data lake could potentially allow for joint collaborations and analysis between various areas, creating shared value within the entire MADE. The benefits of such a project would be substantial. All of this is also valid when applied in other organizations outside MADE. With a single, centralized system, companies can better exploit synergies between data collected in different parts of the company. However, it remains to be understood how to manage a data lake composed of such heterogeneous data from such diverse sources.

2.4. Metadata classification

To understand the usefulness of the different types of metadata, we asked the different area managers to evaluate the metadata classifications of A. Oram and Sawadogo et al. These two classifications differ because Oram's one is focused on the functional aspect of metadata while the one proposed by Sawadogo et al. classifies data based on the structural metadata types. In addition, Sawadogo's classification is very detailed, while the other is broader. Assessing the usefulness of both these classifications allows us to obtain more information and insights about the needs of different areas.

2.4.1. Functional metadata

A. Oram's classification (2015) distinguishes metadata according to how they are gathered. Different types of this metadata category can overlap with each other. Here we find:

- **Business metadata**: these metadata are the semantic annotations that make the data more comprehensible to the end user.
- **Operational metadata**: provide information that can be automatically collected during the data ingestion and processing, which characterizes the data.

- **Technical metadata**: they describe the ingested data's format, structure, or schema.

Table 5 shows some examples for each metadata category.

Table 5. Oram's classification examples

Business	Operational	Technical
names, business terms, business rules, integrity constraints, descriptions	size of the file, location in the system, source, number of records, process information	format (raw text, JPEG image, JSON document, etc.), structure, schema, data's length

2.4.2. Structural metadata

Sawadogo et al. (2019) classification instead calls any data within the lake "object" and considers tree metadata typology that categorizes metadata into intra-object, inter-object and global metadata.

- **Intra-object metadata**: characteristics related to a single object within the repository.
  - **Properties**: a general description of an object
  - **Summaries and previews**: annotations that help to understand the meaning of an object
  - **Semantic metadata**: an overview of the content or structure of an object
- **Inter-object metadata**: represent the relationships existing between the different data.
  - **Object groupings**: organise objects into groups. An element can belong to more groups
  - **Similarity links**: they show how similar two or more objects are based on data intrinsic characteristics

    - **Parenthood links**: reflects that an object can be the union of many others
- **Global metadata**: are data structures intended to give a contextual layer to the entire data lake.
    - **Semantic resources**: are knowledge bases used to generate other metadata and improve analysis
    - **Indexes** are data structures that facilitate quick object searching and discovery in a data lake.
    - **Logs**: are used to track users' interactions with the data lake

Examples are also given for this classification in each category, as shown in Table 6.

Table 6. Sawadogo et al. classification examples

Intra-object metadata		
Properties	**Summaries and preview**	**Semantic metadata**
object title, file name and size, date of last modification, location within the file system	word cloud, data schema, versions and representations,	descriptive tags, textual descriptions
Inter-object metadata		
Object groupings	**Similarity links**	**Parenthood links**
group by tags, business categories, properties, format, language...	refer to intrinsic properties of the object	relationship between source data and structured data
Global metadata		
Semantic resources	**Indexes**	**Logs**
Ontologies, thesauri, dictionaries, taxonomies	keywords, patterns, colours, simple text indexes	log-in records, changes, views

2.5. Metadata classification evaluation

The objective of this phase of the interviews is to understand the usefulness of the different metadata according to the area managers. To do this during the interviews, we explained the classifications of Oram and Sawadogo et al. by asking the

interviewees to rate from 1 to 5 the usefulness of all the different metadata categories. In addition to the grade, respondents were prompted to justify the rating by justifying the usefulness of the metadata in their area of competence. The following chapters summarize the outcome of these interviews.

2.5.1. Oram's classification: functional metadata

Oram's classification was the first to be presented as it is easier and more straightforward to understand. The first comment, shared by most of the managers, was that this classification is very broad and general, making all three categories practically essential for analysis. With the information contained in these metadata, a general profile of the company can be drawn up by understanding what data the company uses and how it uses it for analysis.

2.5.1.1. Business metadata

This category was rated as most useful by most managers. This is due to the polyvalence of the usefulness of this metadata: it is essential for data analysts to search for data within the data lake and at the same time provide a general overview of the analyzed dataset for managers. This dual utility can also reduce the number of errors when the manager requests a certain analysis. With a greater number of tags explaining the data in business terms, analysts will be able to find the data of interest more easily performing accurate analyses. The only outlier in this rating is area 4 because for the analyses they currently perform, there is little need for additional data information since they rely mostly on raw information about the data provenance.

2.5.1.2. Operational metadata

Operational metadata was found to be divisive in terms of utility. These types of metadata are fundamental for any kind of analysis, but many managers have made it clear that they can be more useful from a data analyst's perspective than from a manager's point of view. In spite of this, those who attributed a high level of usefulness

emphasized the importance of the provenance, quality and traceability of the data, fundamental information in areas 2 and 4, for example.

2.5.1.3. Technical metadata

This latter category got less utility than the others. The reason for this is mainly attributable to the fact that in the current situation, knowledge of the structure or schema of the data is of little importance since areas only work with structured data. We do not doubt that in a usage context such as the data lake, where structured, semi-structured and unstructured data are stored, the importance of this metadata will increase.

Table 7. Oram's classification grades by MADE area managers

	Business	Operational	Technical
Area 1	5	2	2
Area 2	4	5	3
Area 3	5	4	3
Area 4	2	5	3
Area 5	5	1	3
Area 6	5	5	5
AVG	4,3	3,7	3,1

2.5.1.4. Conclusion

Oram's classification is often cited in the literature probably because it takes inspiration from the categorization of metadata in traditional databases. Despite this, as explained in Chapter 1.9.1 Diamantini et al. pointed out that different categories of

metadata can overlap, blurring the differences between them and creating confusion for respondents. The use of another metadata classification is therefore necessary to understand which data lake features are required. Table 7 shows the evaluations of all managers regarding this classification

2.5.2. Sawadogo et al. classification: structural metadata

Using this classification for the evaluation of the usefulness of metadata, the respondents showed higher comprehensibility, given by the greater detail of this second classification. This also allowed them to compare the usefulness of the different information associated with the data with the ones currently in use by them (such as PLM product lifecycle management programs). For a more exhaustive evaluation of the categories, respondents were asked to evaluate both the usefulness of the sub-categories (properties, similarity links, indexes...) and the usefulness of the macro-categories (intra, inter, global metadata).

2.5.2.1. Intra-object metadata

According to the answers, intra-object metadata is the starting point on which both inter-object and global metadata are built. These are important because if in a data lake all objects are well-constructed analysis is easier and the other two macro-categories are unimportant if this is not done correctly. This metadata makes it possible to define the content of data, making them useful for both managers and data analysts.

1.1.1.1.1. Properties (PR)

This metadata provides a general description of the data, thus providing useful information to understand what I can get out of that data. The information contained in this metadata is often a starting point for analyses, such as for analyses performed on a particular time frame as often happens in area 4. The usefulness of this information has therefore not been questioned by anyone as there can be no files in the system without this information.

1.1.1.1.2. Summaries and previews (S&P)

The usefulness of this metadata is average. This is because the usefulness of previews of the data content is recognized by the majority of those responsible but in practice, the amount of data currently collected does not justify its use. The high usefulness recorded in areas 2 and 3 was attributed to the usefulness of data versions. In our opinion, summaries and previews become very useful in a context where there are large amounts of unstructured data collected, in order to have an overview of their content. So, in the current situation where only structured data are used the usefulness of these metadata may be out of focus.

1.1.1.1.3. Semantic metadata (SM)

The answers regarding this sub-category returned different points of view. The difference between these depended on whether or not there is a problem with the semantics used within the area. For example, in areas 2 and 6, there were never any problems with the semantics used for the metadata, in area 6 for example was established a policy regarding the semantics to be used, eliminating all the problems that could occur. In area 4, on the other hand, a lot of measurement data is saved manually by the operator, which causes problems when searching for data slowing down it when different terminologies are used. When searching for data within the repository they often search by date and not by name. Area 1 manager stated that they also encounter numerous semantic problems when dealing with external information system providers. Different companies use different vocabularies and these differences are also present within the same company. Summing up the effectiveness of semantic metadata depends on two main factors:

- Intra-company factor: the presence of regulatory policy regarding the semantics to be used within the company reduce problems with semantic metadata and increases their effectiveness.

- Extra-company factor: the importance of semantic metadata increases considerably when dealing with external parties since it helps to understand the data meaning as long as are used the same terminologies in both companies.

Table 8 shows the ratings of all managers regarding these metadata.

Table 8. Sawadogo's intra-object metadata classification grades by MADE area managers

	Properties	Summaries and previews	Semantic metadata
Area 1	4	3	5
Area 2	4	4	1
Area 3	5	4	4
Area 4	5	3	5
Area 5	5	2	3
Area 6	5	3	3
AVG	4,7	3,2	3,5
	INTRA-OBJECT METADATA: 3.8		

2.5.2.2. Inter-object metadata

The usefulness of this macro category strictly depends on the goodness of the intra-object metadata. Inter-object metadata is useful for correlating data within the data lake, improving search and facilitating the discovery of data clusters for analysis. The importance of this metadata is directly proportional to the amount of data collected. For example, for predictive maintenance, you need as much information as possible about the state (temperature, acceleration...) of a machine. In area 4 instead, it is essential to associate products with the different processes to make an analysis. This

data should then be linked together to know the machine's state and process state, making data relationships very important. Table 9 resumes the evaluations of all managers regarding these metadata.

1.1.1.1.4. Object groupings (OG)

This function represents the basic function of inter-object metadata. It represents the 3rd sub-category considered most useful by the managers of the different areas with the exception of the 'Virtual Design and Product Development' area, which considered the other categories more important than this one for their research. Obviously, the function of associating metadata based on semantics turns out to be a widely used way of searching data, if not even necessary.

1.1.1.1.5. Similarity links (SL)

Similarity links metadata can be considered an advanced function compared to the other two inter-object categories. To integrate them into the data lake, the development or implementation of plug-ins is required to analyze texts, photos, etc. to self-generate metadata. In area 3 could be interesting to see links automatically generated from different data sources, as it is not always obvious. Area 1 manager said he was very interested in the opportunities this function could offer for identifying similarities between their data. In predictive maintenance, however, given the amount of data to be considered at the same time, this function is not essential compared to object grouping. We can conclude that this metadata is always considered to have high potential but in some contexts more than in others.

1.1.1.1.6. Parenthood links (PL)

This function shows a very high degree of usefulness. This is due to the fact that many areas move within their PLM software precisely with tools that hierarchically display products and documents. The grouping of elements that belong to the same product, process or source data makes it much easier to understand the data. The definition of the hierarchy of data can also be useful for the presentation of data according to the

user or to assign access rights. In area 2 and in FMS where there are many products flows it can be very useful or even used for BOMs for keeping track of what is added to a product. In area 3, on the other hand, more importance is given to these metadata in order to keep track of the evolution of the analysis and modification of the data. Important in this category is to define the right level of granularity to help data understanding, even visually, without making it more complicated.

Table 9. Sawadogo's inter-object metadata classification grades by MADE area managers

	Object groupings	Similarity links	Parenthood links
Area 1	1	4	5
Area 2	4	1	5
Area 3	5	3	4
Area 4	4	2	5
Area 5	5	1	4
Area 6	5	1	3
AVG	4	2	4,3
	INTER-OBJECT METADATA: 3.45		

2.5.2.3. Global metadata

Global metadata is data structures that apply to the entire data lake to keep track of interactions and to facilitate searching. Within MADE, the usefulness of this metadata is recognised especially when the number of data increases and when there are exchanges of information between areas or with third parties. Table 10 shows the managers' evaluations regarding these metadata.

1.1.1.1.7. Semantic resources & Indexes (SR) & (IX)

These two categories of metadata allow to easily find data in a similar but opposite way. Semantic resources allow additional tags similar to those already present, to be attached to the data. This facilitates the discovery of the data, as the semantic tag searched is more likely to be associated with the file. On the other hand, with indexes, in the moment in which a user queries the data lake with a textual query, the system will also search for semantic tags similar to those entered by the user. The function is the same, but executed differently and with different benefits:

- Semantic resources: allows multiple metadata tags to be permanently associated with the data. In this way, if the data leaves the organisation's data lake, it will still have numerous data tags that will allow even 3rd party systems to easily identify it.
- Indexes: By not attaching new tags to the data, these are not weighed down in terms of file size. On the other hand, in the moment where the data leaves the organisation's data lake, indexing on that data will no longer work.

Since most queries are coded by name or code, these tools are particularly useful with a comparable average utility. Areas that collect less data express low usefulness for this metadata as far as the current situation is concerned. But, as more data are collected these became important, as well as increasing data exchanges between areas.

1.1.1.1.8. Logs (LS)

This metadata is important for identifying responsibility for modifying or updating data. For example, in area 3 when software or sensors report problems, it is important to understand who modified the system. For many areas, moreover, data security has not been declared a priority, except for area 5 since working with 3rd party data, privacy is important.

Table 10. Sawadogo's global metadata classification grades by MADE area managers

	Semantic resources	Indexes	Logs
Area 1	5	3	2
Area 2	2	3	2
Area 3	3	5	3
Area 4	4	2	2
Area 5	2	3	5
Area 6	3	4	2
AVG	3,2	3,3	2,7
	GLOBAL METADATA: 3		

2.5.3. Conclusions

As explained, Oram's classification is much more generic than the one of Sawadogo et al. In addition to this, the structural classification includes all the functionalities of Oram's classification. Business metadata are practically encompassed in semantic metadata, operational metadata are comparable to logs and other inter-object metadata, while operational metadata fall under preview and summaries metadata. The structural classification can thus be regarded as an evolution of the functional classification, making it more optimal for evaluating the functionality required in the data lake.

This is also confirmed by the analysis of P. Sawadogo and J. Darmont (2021), where they investigate the similarities between the different classifications as shown in Table 11. However, the use of both classifications for interviews proved to be useful, to collect more data and hear more opinions.

Table 11. Similarities between Functional and Structural metadata

	Functional metadata	**Structural metadata**
Basic characteristics of data (size, format, etc.)	***X***	***X***
Data semantics (tags, descriptions, etc.)	***X***	***X***
Data history	***X***	***X***
Data linkage		***X***
User interactions		***X***

Starting from Table 12 we will now group structural metadata into 3 macro utility groups, basing on the mark that every single metadata received. We distinguish between essential, useful, and advanced metadata.

Table 12. Final metadata ratings by MADE area managers

	PR	S&P	SM	OG	SL	PL	SR	IX	LS
Area 1	4	3	5	1	4	5	5	3	2
Area 2	4	4	1	4	1	5	2	3	2
Area 3	5	4	4	5	3	4	2	5	3
Area 4	5	3	5	4	2	5	4	2	2
Area 5	5	2	3	5	1	4	2	3	5
Area 6	5	3	3	5	1	3	3	4	2
AVG	4,7	3,2	3,5	4	2	4,3	3	3,3	2,7
	INTRA: 3.8			**INTER: 3.5**			**GLOBAL 3**		

2.5.3.1. Essentials metadata

In this category we find properties, object groupings, and parenthood relationships. With these metadata, it is possible to describe an object with basic metadata, group it based on these attributes, and keep track of the relationships and hierarchical structure of the data. Properties and object grouping provide basic functions for any database. The usefulness of this metadata is that many area managers work using this information. Particularly in the case of parenthood relationships applied to the data catalogue, they greatly facilitate understanding and navigation within the system. Without this information then, it is very difficult to interface with the data and understand the content of the repository.

2.5.3.2. Useful metadata

Here we find instead the remaining intra-object metadata, summaries & preview and semantic metadata, together with indexes. The semantics used for metadata is often causing confusion and slow down the analysis process. These metadata, if not essential for the proper functioning of the data lake, prove to be very useful especially when there are no policies for semantic standardisation within the company. In the context of MADE indexes have proven to be more useful than semantic enrichment, this is true as long as the exchange of data outside the organisation is not high. Concerning summaries and previews, those responsible did not attribute high usefulness but recognised that in a context where unstructured data are also collected, this metadata increases their utility significantly.

2.5.3.3. Advanced metadata

In this last category, we find metadata that enables useful functionality when large amounts of data are collected or when privacy is of key importance. With "advanced" we mean functions that are considered something more than basic metadata management operations. Similarity links allow the aggregation of data based on common characteristics intrinsic to their content, a useful metadata for the automatic aggregation of data for future analysis. Logs are not essential for performing analyses and in the current context of analysis they are not recognised as useful. Finally, as far as semantic resources are concerned, they become important when the amount of data and the exchange of data with 3rd parties external to the company is large. Table 13 distinguishes between essential, useful and advanced metadata.

Table 13 Essential, Useful, and Advanced metadata

Essential metadata	Properties
	Object groupings
	Parenthood relationships
Useful metadata	Summaries and previews
	Semantic metadata
	Indexes
Advanced metadata	Similarity links
	Semantic resources
	Logs

3 Relationship between metadata and features

3.1. Intro

As emerged in the discussion introduced in the previous chapter, metadata are crucial to properly manage data in a data lake. Nevertheless, not all the metadata are always needed. Conversely, there are metadata which are fundamental to support some data lake features and less useful for other feature. For this reason, to define which metadata model will be optimal within I4.0 environments, the information gathered during the interviews with the different stakeholders are also useful to determine which are the relevant features and to link the metadata categories with the features they enable.

3.2. Data lakes features

Features provided by a data lake have been identified and classified by Sawadogo et al. (2019) and R. Eichler et al. (2020) with two distinct classifications.

Sawadogo et al. (2019) distinguish between six main features:

- **Semantic Enrichment**: enable to add textual descriptions to data, describing their content and making the data more comprehensible.
- **Data indexing**: this is the search engine function of a data lake. It allows searching for data using keywords or patterns making it easier to search for data within the data lake.

- **Link generation and conservation**: this function makes it possible to generate links between different data in order to facilitate searching. These links can be generated manually or automatically by the tools.
- **Data polymorphism**: if data is transformed to be adapted to a new context, there must be a reverse function that allows going back to the original state. In this way, multiple representations of the same data are allowed at different levels of detail or structure. It refers to the fact that I can find the same data more or less structured, altered, or adapted depending on the context.
- **Data versioning**: this function makes it possible to update or modify data while maintaining previous states. This functionality automatically relates two or more data, one of which is the latest updated or modified version of the other.
- **Usage tracking**: allows the recording of iterations (creation, access, and update of data) between users and the data lake.

R. Eichler et al. (2020) classification instead identify:

- **Metadata Properties**: you need to have metadata useful for flexible lake management. So, you need to have information about the semantics, properties, and relationships that the data have.
- **Granularity Levels**: it is necessary to consider that data can be aggregated hierarchically following the various dimensions, according to the context where this data belongs. For example, a KPI can be presented for a single machine or aggregated on the entire shop floor.
- **Data Zones identification**: data lake users need to have information about the position of the data in the architecture and thus keep track of the path it has taken. It is needed to understand how much the data item has been altered to fit the context.

- **Categorization**: labels to identify the category of a piece of data that then make it findable by the user.

To search for the best classification of features required for proper data lake operation, the following section sets out to compare the work of Eichler and Sawadogo to see if there are conceptual holes on either side and to look for any similarities.

3.2.1. Metadata Properties vs. Semantic Enrichment/Link Generation

Metadata Properties is a feature also described by Sawadogo. According to its feature classification, Sawadogo, describes Eichler's Metadata properties in two distinct parts: Link generation and semantic enrichment. So, what is described by Eichler is unpacked more clearly by Sawadogo. Indeed, recall that Eichler described metadata properties as information about semantics, properties, and the relationships that data have with each other. In our opinion, Sawadogo's choice is clearer since link generation and semantic enrichment are two different functionalities and it is right that they should be considered separately. Semantic enrichment, on one hand, involves semantic and properties information to make the mere data more understandable. Link generation, on the other hand, focuses more on generating links by correlating data based on intrinsic features or properties. Therefore, a classification in which these two features are considered separately is preferred.

3.2.2. Data Zones Identification vs. Data Polymorphism

We also find similarities between data zones identification and data polymorphism. The first one is defined as the need to have information about the position of the data in the architecture, keeping track of the various versions of the data, altered according to the context. It is thus similar to data polymorphism, although data zones also require tools about the geographic location and the path the data takes to be fully satisfied as a functionality. We believe that the two functionalities are similar and so data polymorphism will be kept in the overall functionalities set.

3.2.3. Granularity Levels

Granularity levels proposed by Eichler, on the other hand, is a feature not covered by the Sawadogo classification. It differs from data versioning in that the latter is more concerned with managing changes and updates on data, keeping track of previous states. It also differs from data polymorphism, in that it refers to the fact that one can find the same data more or less structured, altered or adapted depending on the context. Granularity levels means keeping track of and tracing back to each level of the hierarchical aggregation of data according to the specific dimension considered. An aspect, therefore, that is missing in Sawadogo's proposed functionalities and that is reasonable to add in integrated classification, since it would allow the data scientist to save time by eliminating repetitive operations, facilitating the management of hierarchies and nested attributes.

3.2.4. Categorization vs. Indexing

Reasoning about the fourth feature proposed by Eichler, categorization, we note that there is a strong similarity with Indexing proposed by Sawadogo. According to Eichler, categorization consists of labels and keywords used to identify the category to which a piece of data belongs so that it can be easily recognized and found by the user because of it. Indexing, proposed by Sawadogo, is exactly the same thing: the use of keywords, patterns or tags to search for data more easily based on the category it belongs to and beyond. We, therefore, believe that only one of these functionalities can be considered two as they are extremely similar.

3.2.5. Data Provenance feature

After combining and analyzing the two classifications, we notice that there is a gray area not covered by any functionality: data provenance. Indeed, we need a function that stores and allows us to trace back any kind of transformation that the data has undergone. This same function must also allow us to trace the physical, geographic

provenance and the path the data has taken to get to the destination where we find it; for example, which machine it came from, whether it came from external data sets or whether it passed through different data zones within the data lake. It is not enough to be able to relate two data sets that express the same information but adapted to each other's context as in the case of data polymorphism. It is not enough to be able to relate data of which one is the updated or modified version of the previous one, as in the case of data versioning. Finally, it is also not enough just to have a function that connects data with their other hierarchical representations, such as granularity levels. We need an additional feature to measure, store and provide quantitative information about the transformations as such. Consequently, we then need to trace back to the physical, geographical and path provenance information of the data.

3.3. Metadata as an enabler for data lake features

Analyzing the data lake features of Eichler and Sawadogo allowed us to identify how to express all the data lake functionalities. Taking Sawadogo's unmodified categorization as a starting point, we added granularity levels and data provenance thanks to Eichler's contribution. Table 14 summarizes the identified features of a data lake.

Table 14. Identified Data Lakes Features

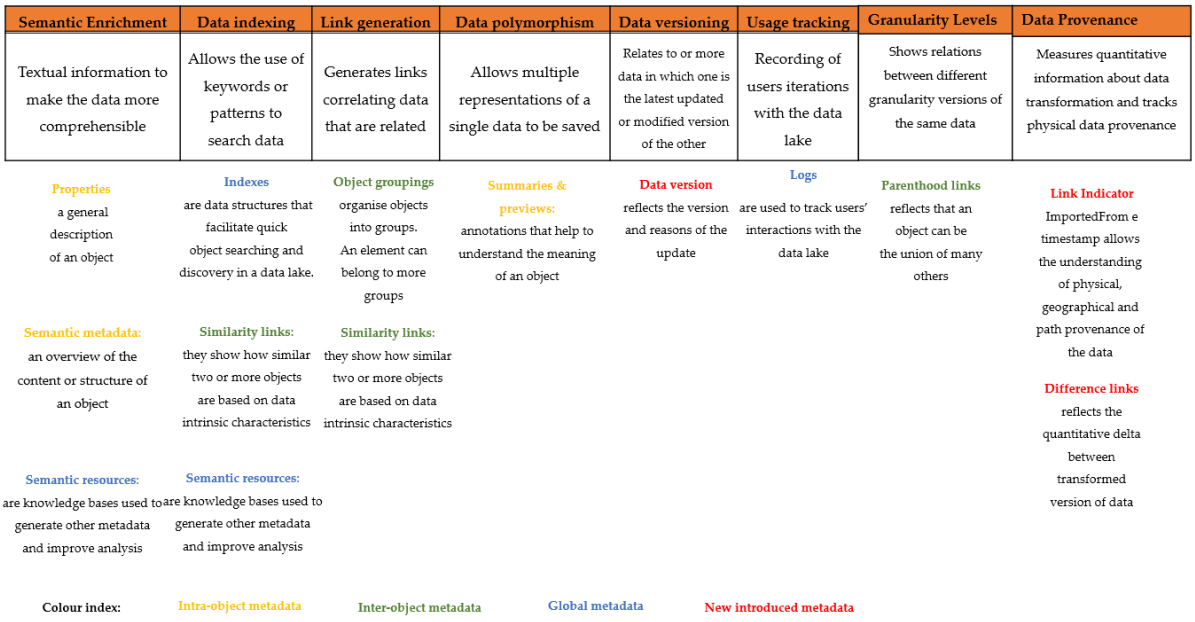
Semantic Enrichment	Textual information to make the data more comprehensible
Data indexing	Allows the use of keywords or patterns to search data
Link generation	Generates links correlating data that are related
Data polymorphism	Allows multiple representations of a single data to be saved

Data versioning	Relates two or more data, in which one is the latest updated or modified version of the other.
Usage tracking	Recording of user's iterations with the data lake
Granularity levels	Shows relations between different granularity versions of the same data.
Data provenance	Measures quantitative information about data transformation and tracks physical data provenance

Since metadata are considered as input for the functioning of a feature, they should be linked according to the feature they enable. This relationship is not exclusive: a feature may be enabled (partially or fully) by more metadata categories at the same time. This does not mean that the metadata are the same, but that a feature may be executed in a different way. Once it is clear how the different metadata and features are connected, the votes on the usefulness of the metadata will be turned to understand the necessity of the different features. This will allow us to identify the metadata model more suited for MADE.

The work done will allow us to link the data lake features with the respective enabling metadata categories. This requires the introduction of three more metadata categories: Data version, Difference links, and Link indicators to enable those features that require new metadata to function. Figure 3 summarizes the results obtained in the following sections.

Figure 3. Data Lakes Functionalities and Enabling Metadata



3.3.1. Semantic Enrichment

Three categories of metadata fall into this category: properties, semantic metadata, and semantic resources. The first two metadata gives information such as the title, data size and all the information specific to the data. This metadata thus makes it possible to know the content of a piece of data within a data lake without the need to open it. Enablers of semantic enrichment are also semantic resources. These are data structures applied to the entire data lake that allows the data to be enriched with metadata relevant to it. Thanks to this metadata, a data item with an already associated semantic tag, will also be added with semantics tags with the same or similar meaning. For example, if we have data with the tag "IoT", the tags "Internet of Things", "Sensor", or "Real-time" will also be added, depending on the corporate ontology used. This makes it possible to associate additional information to the data in order to better understand its content.

3.3.2. Data indexing

Three metadata of those presented enable data indexing functioning: indexes, semantic resources, and similarity links. Indexes is present exclusively as it does not enable other functions. Indexes are data structures that make it easy to search for data within the data lake, functioning as a search engine. Thanks to this metadata, the user can query the data lake and find data with semantic tags different but related to those searched. For example, when a user wants to search for data related to "predictive maintenance", even though there is no data directly related to the concept of preventive maintenance in the system. With specially developed indexes, the system will return data with the associated tag "Area 5", "Monitoring", or "Failure". Indexes can also be used to search using images, audio and video.

Enabler of this function is also semantic resources and similarity links. These metadata also enable other functions, since their main purpose is not to facilitate searching, but indirectly they greatly facilitate it. Semantic metadata enriches data with additional metadata, which will then appear in searches, as well as similarity links that automatically relate similar data for future searches.

3.3.3. Link generation

Within this feature, we find two inter-object metadata: object grouping and similarity links metadata. Object grouping metadata allows data with the same metadata tags to be related, building the basis for the analysis of aggregated data. Similarity links, on the other hand, allow different data to be automatically linked on the basis of their intrinsic content. For example, if English and Italian text documents are loaded into the data lake, the system can analyze the document, understand whether it was written in English or Italian, and associate a tag that makes the language explicit, relating them. This metadata can be associated not only based on text similarities but also based on patterns, data series and other inherent characteristics of the data

3.3.4. Data polymorphism

Data polymorphism is enabled by Summaries & Previews. Summaries and previews often result to be summary files that are in any case less detailed than the source data. For example, excel files made up of several columns can be measured by considering different units of measurement. This function can also be used for privacy purposes. Different versions of the same file may be created to hide sensitive information from users who do not have access rights. Another case is when unstructured data is transformed into structured data. This resulting file is still metadata relative to the original file as it provides the same information but is shown in a different way. These metadata can be considered as versions of the original files, showing the same data in a diverse way, with different units of measurement, or structure. The key concept in data polymorphism is that starting from any version of the data, it is then possible to go back to the original version.

3.3.5. Data versioning

There is no metadata in the Sawadogo classification to enable this function. When a file is updated within the system, it is important to have information that relates the file to the original one, such as the file version and the reason for the update. Without this information, there is a risk of confusion within the repository, between files that may also be very similar to each other. It is therefore necessary to enrich Sawadogo's classification with another category of metadata showing this information.

3.3.5.1. Data version metadata

This metadata reflects the version and reasons for a data update. The identification of the most up-to-date file within the system is crucial in order to perform data analysis on the correct files. Information about the reasons behind an update, on the other hand, is useful to distinguish updates of the file content from those due to errors in data collection. A file that is updated several times does not automatically mean that it is a

problem, but to understand this we need information on the reason for the update. This metadata falls under inter-object metadata as it relates to two distinct data elements within the data lake. This information may be generated automatically, as for versions, but also manually to annotate the reasons for updating.

3.3.6. Usage tracking

Logs turn out to be the only metadata required to enable this function. These metadata enable the recording of data access by various users. By linking the accountable party, it is possible to track the history of accesses and updates. This data can be used to determine which data has been updated or utilized the most frequently.

3.3.7. Granularity Levels

Representing data at different levels of detail was found to be very important in the interviews. Parenthood links metadata are needed to enable this function. They fall under this feature as these allow the original data to be seen as a subset of smaller data. Unstructured data can be represented as a union of structured data, or if we think of a product, this can be represented as the union of its component parts. With excel files made up of several columns, a second summary file can be created to highlight only some columns and aggregate others. For example, a KPI can be measured by considering longer or shorter time intervals or considering more or less machinery aggregated. Especially for searching data within the system, function and the related metadata are very useful. Files located at a lower level of the hierarchical structure are considered as metadata of the original file that provides a picture of the data contained in a more detailed way than the properties metadata. Therefore, it will be of paramount importance to establish hierarchies among the different data, to facilitate their organization and thus the understanding.

3.3.8. Data provenance

Regarding this data lake feature, there is no direct correlation with the metadata classification of Sawadogo. This is because there is no metadata in the classification that gives information about the differences between two files in the data lake. This information is important to understand the differences, the reasons, and the changes of an update. This information, while not immediately useful for performing analysis, is very important for understanding the differences between multiple versions of data. It allows to understand which kind of transformation has been made to the data for assessing the change per se in a quantitative way. The other conceptual hole filled by data provenance is keeping track of information about where the data came from physically, geographically, and the route the data took to get to the destination where we find it.

What we propose then, is the introduction of other categories of metadata to be added to Sawadogo's.

3.3.8.1. Difference links metadata

The first is called Difference links and are the metadata that is generated when updating data within the repository. This metadata contains information regarding what distinguishes it from the previous versions. So, it will provide information regarding the items involved in the update, the reasons for it, indicators of change, and information about the delta of the versions. All this information can be extracted manually (in the case of a one-time update) or assigned automatically by integrating additional functions. Since in the concept of data versioning different versions of files are maintained separately, this new metadata is to be considered as inter-object metadata since they link different data within the data lake.

3.3.8.2. Link indicators metadata

Link indicators metadata stores the information about the source from which the data was imported into the zone as well as the appropriate timestamp. The name of the original source or a zone may be included in the importedFrom attribute. The connection and importedFrom attributes make it possible to follow the passage of the data through the zones, following the route that data followed to arrive at the destination where we find it. This type of metadata will be an intra-object metadata since will be referred to only one data object in the lake.

4 Data lake features evaluation in I4.0

Once we have defined the relationship between data lake features and metadata categories, we need to understand the utility of different features in IoT and I4.0 environments. This will then allow us to select the most appropriate metamodel based on the needs of I4.0 related companies. In this section, we are going to look at what data lake features are most useful at MADE. As explained in the previous chapter, metadata serves as input for the proper functioning of the features. During the interviews, the usefulness of the different features was not asked but only those related to metadata. This was done to make it easier for area managers to understand metadata classifications but also to avoid that more utility was attributed to desired features rather than those that are effective. This is because managers are well versed in the concept of metadata and therefore aware of its usefulness, whereas asking about the usefulness of different features might have favored what managers desire over what they need. In this way, however, we have no votes regarding the usefulness of the features, an essential requirement for being able to select the most correct metadata model. It will then be necessary to define a way to turn the utility of the different metadata on the data lake features enabled.

4.1. Grading New Metadata Types

Based on the interviews done although we have no quantitative information regarding the votes for the three new metadata, but we can state that there are some preferences, both from MADE area managers and data scientists, regarding the new metadata introduced earlier. In fact, in the interviews done with the data scientists, conceptual gaps emerged in the set of metadata presented to them and thus they indirectly disclosed the importance of new metadata categories. Area managers, as discussed extensively in the interview chapter, gave insights with less technical information

related to the daily use of data in their area. Based on this information gathered in the interviews, we will put the three new metadata added to the set into the categories Essential, Useful, and Advanced. We will then associate the new metadata with the average rating of each category they belong to get a quantitative idea of their importance. The categories are, in descending order of importance, essentials metadata, useful metadata and advanced metadata. The essentials metadata we recall being indispensable for the management of any data lake, as they greatly facilitate even the most basic use of data. Useful metadata are potentially useful but not yet fully utilized in MADE. Advanced metadata, on the other hand, would allow full use and management of the data lake, making it possible to query the data in an advanced way because of the power these tools have.

Let us now turn to the assessments established for the three new metadata, which we recall are Difference Links, Data Version, and Link Indicator. Due to their nature, in the following, the first two are analyzed together, while the third as a separated metadata.

4.1.1. Difference Links & Data Version metadata evaluation

Difference Links and Data Version share a common fate in that both have been associated with the category of advanced metadata. These metadata are concerned with keeping track of changes and updates in a quantitative manner made to the data and associating the various versions of the same data with each other. Both, therefore, would be very useful in a dynamic context, where the same data is frequently modified and updated. They are metadata useful for managing a more advanced level of data lake, thus not essential or useful to the MADE as-is context, since neither the processes that generate data nor the processes that manage the data require such meta-information regarding the changes undergone by a data object. So, making use of the information gleaned from the interviews with area managers, we were able to associate Difference Links and Data Version with the advanced metadata category and

give it a rating of 2.56, corresponding to the average rating of the metadata previously associated with that category.

4.1.2. Link Indicator metadata evaluation

Link Indicator is considered indispensable for lake management within MADE, so we associated them with the essentials metadata category. In fact, from the interviews with data scientists, it became apparent that the classification presented to them lacked a fundamental functionality regarding data provenance. Consequent to this they also lacked a tool to keep track of what physical source, what database, or what area that data came from, to trace the provenance and be able to contextualize it. It is here, then, that Link Indicator plays an indispensable role within MADE as-is data management. Based on what the area managers said, it was decided to add Link Indicator to the essentials metadata category and associate it with the average rating of 4.33. Table 15 groups the final set of metadata needed.

Table 15. Complete Set of Enabling Metadata

<table>
<tr><th>Essential metadata</th><th>Useful metadata</th><th>Advanced metadata</th></tr>
<tr><td>Object groupings</td><td>Summaries and previews</td><td>Similarity links</td></tr>
<tr><td>Parenthood relationships</td><td>Semantic metadata</td><td>Semantic resources</td></tr>
<tr><td>Properties</td><td rowspan="3">Indexes</td><td>Logs</td></tr>
<tr><td rowspan="2">Link indicator</td><td>Difference links</td></tr>
<tr><td>Data version</td></tr>
</table>

4.2. Utility conversion method

At this point multiple methods were discussed and analyzed to select the method that most closely aligns the usefulness of metadata, the comments made on it, the responses regarding current data usage, and data lake features. The first thing that has been established is the presence of a directly proportional relationship between the utility of metadata and the respective feature enabled. This is because if a metadata category is considered useful, the function (and the information) it enables is also useful.

The main issue is that certain data lake features are enabled by multiple metadata categories that belong to different utility categories (essential, useful, advanced) and thus have different ratings. This at first did not allow us to report the utility of metadata directly on the respective feature. The first solution considered was to average the enabling metadata ratings for each feature as shown in formula:

$$Data\ lake\ feature\ utility = \frac{\sum enabling\ metadata\ utility}{n°\ of\ enabling\ metadata}$$

Table 16 summarizes the feature's utility using this formula.

Table 16. Utility Grades with Average Method

Features	Enabling metadata			Utility
Semantic enrichment	Properties: 4,7	Semantic metadata: 3,5	Semantic resources: 3	**3,7**
Data indexing	Indexes: 3,3	Similarity links: 2	Semantic resources: 3	**2,8**
Link generation	Object groupings: 4	Similarity links: 2	///	**3**
Data polymorphism	Summaries & previews: 3,2	///	///	**3,2**
Data versioning	Data version: 2,7	///	///	**2,7**
Usage tracking	Logs: 2,7	///	///	**2,7**
Granularity levels	Parenthood Links: 4,3	///	///	**4,3**
Data provenance	Link indicator: 4,3	Difference Links: 2,7	///	**3,5**

This method, although logical, has a major problem as it does not respect the principle of direct proportionality between metadata and features. As shown in Table 16 the

total utility of Semantic enrichment is calculated as the average of the utility of Properties, Semantic metadata and semantic resources. Considering the comments of area managers and since the utility of Properties is the highest among all categories, it is expected that this feature will also present a high rating. Using this method, however, this is not the case because the other metadata categories have a rating that lowers the average. In this example, there are also other features that get a higher final rating as highlighted in the table, such as Granularity levels. This from a conceptual point of view is incorrect because the three metadata categories are independent from each other. So, a low utility of one should not lead to a lowering of the rating of the feature that can enable all three individually. This problem leads to lower ratings of features that are enabled by multiple metadata simultaneously.

To solve this problem, it was also considered to use a weighted average that gave more importance to metadata that appeared exclusively in a category (in the case explained before properties), but the weight assignment turned out to be arbitrary and it would not have completely solved the problem.

In the end, it was decided to assign to each feature the higher utility of the metadata that uniquely enables the function as shown in formula:

$$Data\ lake\ feature\ utility = MAX(enabling\ metadata\ utility)$$

This is the method that best converts utilities according to the evaluations and the discussions held with different area managers. Table 17 shows the data lake features utility final evaluation.

Table 17. Utility Grades with Maximum Enabling Metadata Method

Features	Enabling metadata			Utility
Semantic enrichment	Properties: 4,7	Semantic metadata: 3,5	Semantic resources: 3	**4,7**
Data indexing	Indexes: 3,3	Similarity links: 2	Semantic resources: 3	**3.3**
Link generation	Object groupings: 4	Similarity links: 2	///	**4**
Data polymorphism	Summaries & previews: 3,2	///	///	**3,2**
Data versioning	Data version: 2,7	///	///	**2,7**
Usage tracking	Logs: 2,7	///	///	**2,7**
Granularity levels	Parenthood Links: 4,3	///	///	**4,3**
Data provenance	Link indicator: 4,3	Difference Links: 2,7	///	**4.3**

4.2.1. Semantic enrichment

This functionality is of paramount importance for a data lake in an industry 4.0 environment, as it provides a quick and easy way to get a clear idea of the domain of the data set. Thanks to tags and semantic information, important information is given to those interfacing with the data lake for the first time so that they can easily understand what they are dealing with. You can understand, for example, what kind of data you are looking at thanks to the tag, you can understand when it was collected thanks to the properties, and you can have the general information, taxonomies, and knowledge bases about the entire set, that allow you to navigate within it, thanks to semantic resources. This makes it possible, very quickly and efficiently, to transfer the knowledge of an area manager to someone who has no knowledge about the domain in question; if a manager, for example of quality control, enriches the data set with the enabling tools of this functionality (tags, semantic resources i.e.), a client can immediately understand the data thanks to these prior knowledge bases that the manager has transferred to the data set.

4.2.2. Data indexing

Without indexing, it would be more complex to find data within the lake. Quickly and efficiently, it allows data to be found through tags, keywords, patterns, or similarities with other data, which, by recognizing the properties of the data, brings it to the attention of those who want to query the data lake. They have a medium-level rating for usefulness, but they are facilitators of an operation that would not be possible without them. Thus, it is an indispensable function for the data lake. As the number of data increases, it becomes increasingly important, as it would become very confusing to search for data without the help of this functionality. What is more, if the Semantic Enrichment part is well structured, the usefulness of Indexing increases proportionally, since if the data is sufficiently categorized through tags, keywords, patterns, or semantic information, it becomes much more powerful the functionality in question.

4.2.3. Link Generation

This feature allows data objects to be associated based on both semantic and structural similarities. It is considered one of the most useful by the results of our interviews. The potential that similarity links have is absolute according to managers in various areas. In an industry 4.0 context where there is a lot of data, where it is often collected from different parts and business functions, and often merged with data from outside the company as well, it becomes critical to have a horizontal view of the various data silos. Specifically, being able to link data from different data sets, which in an analytics context without a data lake, would never have come together, is considered very useful. Link generation thus allows a horizontal rather than a vertical view of different data sets. For example, one relates data on the quality of a component, for example, with downtime caused in its processing or with feedback received from customers regarding problems encountered in the use of the final product. What is made possible by link generation is a three-hundred-and-sixty-degree analysis of the knowledge contained within the lake.

4.2.4. Data polymorphism

This feature makes it possible to go back to previous versions of any data, in case the data has undergone updates or changes to fit the context it is in. In this case, the higher the complexity of the lake, the more sources from which the data is drawn, the more linked data sets that make up the entire lake, and the more important data polymorphism as a feature takes on. Indeed, recognizing that it is the same data, transformed and altered to fit a different context, becomes important in order to perform complete and correct analyses that do not include duplicates. The inverse function of the data, which allows it to be traced back to its original version, turns out to be crucial to check if errors were made in the modification, thus validating the analyses and reducing the GIGO (garbage in garbage out) effect. Thus, it becomes important once the database becomes complex and large; it is therefore not essential

to the operation of a data lake, but it grows in usefulness proportionally to the data lake, becoming a useful facilitator.

4.2.5. Data versioning

This feature is used to relate different versions of the same data, since it can be updated, it is also used to understand the reason for the update. For example, if I am looking for a KPI to conduct an analysis, I will need to find and know the most up-to-date version and maybe evaluate it against earlier versions as well. This is an advanced function, which, as far as our surveys are concerned, is not fundamental to the management and proper use of a data lake. It can be useful in dynamic contexts, where data undergo many changes and updates. In a context such as MADE, it would turn out not to be fundamental for the time being, as area managers state that they do not perform numerous updates on the data.

4.2.6. Usage tracking

This data lake feature allows using metadata to track users' accesses and the interaction that they make with the data lake. This one is not fundamental for proper data lake management. If there are many users and the security of some data is considered a priority, then the importance of usage tracking increases. In addition, if you work with partners outside the company, usage tracking is likely to become important, as there may be data leaks or errors in the data lake. It is therefore considered an advanced feature.

4.2.7. Granularity levels

This functionality is important because it introduces the concept of hierarchies, as it shows, allows one to trace and move along the family tree of a data item. In fact, a datum can be the union of two others, or, for example, a KPI can be represented for a single machine or refer to the entire corporate shop floor. Granularity levels are therefore the only way to consider and recognize the level of aggregation of a data object, it, therefore, becomes essential to avoid, for example, doing analysis by

considering data from different hierarchical dimensions. It also allows one to trace back to errors, by disaggregating a KPI, and thus recognizing all the micro components with which it is calculated, one can trace back to the underlying problem and solve it. It could be that in a production line there are frequent delays, given by downtime: with a KPI generic to the line I notice delays, but by unpacking this KPI at each station one can see where the problem is located. From surveys conducted with MADE managers, it, turns out to be a key feature for the data lake.

4.2.8. Data provenance

This feature allows one to keep track of the path a data item takes before arriving at the location where it is; it allows one to recognize the path the data item took, going all the way back to the database or from the machine from which it was extracted or generated. It is also useful for recognizing quantitative differences in a data item from its previous version; for example, if a data item is updated weekly, such as the number of deliveries processed may be, I can understand the difference there is between the current week and the previous week. It, therefore, gives the opportunity to move nimbly within the lake, extracting information that would otherwise be unreachable. This is why it is considered an essential feature for the proper management of a data lake.

4.2.9. Conclusion

To conclude the discussion regarding the usefulness of the features we have divided them into two categories: advanced features and basic features. The basic ones are functions that are indispensable for the operation of a data lake in I4.0 industries, functions from which one cannot disregard; without them, the data lake would turn into a data swamp. Advanced features, on the other hand, increase in usefulness as the complexity of the lake increases. By complexity we mean, a large number of data, extracted from numerous and heterogeneous sources, or a dynamic context, in which data are updated frequently. Referring then to the Industry 4.0 context, the table 18

resumes the feature set chosen for the proper management and maintenance of a data lake. The only issue is the level of complexity and maturity of the data lake, which if low, can be managed only with the basics while as data saved increases the more advanced features will become indispensable.

Table 18. Data Lakes Basic and Advanced Features

Basic Features	Advanced Features
Semantic enrichment	**Data polymorphism**
Link generation	**Data versioning**
Granularity levels	**Usage tracking**
Data provenance	
Data indexing	

4.3. Interviews with industry experts

To validate the results, two interviews were held with experts in the field. Notably, Enrica Bosani of Whirlpool and Alberto Erisimo and Conte Giovanni from SAP have been contacted. The companies they work for have implemented some use cases in one or more areas of MADE, making them perfect for obtaining new insights about the work done. Specifically, the interviewees were presented with the metadata categories and data lake features to comment on their usefulness, always taking into

consideration the MADE and I4.0 environments. During the conversations, the current state of data management in companies and related problems were also discussed.

4.3.1. Enrica Bosani – Whirlpool

Enrica Bosani is an electrical engineer working in industrial automation and coordinator of Industry 4.0 projects for Whirlpool. They work exclusively with structured data saved in the cloud where no such elaborate structuring of the data is done as the one presented by us in Chapter 4.1.2.

Concerning metadata categories, a lot of importance has been attached to Summaries & preview metadata, which are used a lot and taken for granted. Considerable importance is therefore attached to intra-object metadata, while inter-object and global metadata are not associated with data at Whirlpool. Although these latter metadata are very interesting, they are not used, as the main function of their databases is the repository. E. Bosani still recognizes the usefulness of this additional information, but they are considered as "advanced" metadata for the function they enable. In terms of potential, it is extremely interesting because it opens up a logic that could turn the analysis of this industry and the technologies to implement them already exist.

From an industrial point of view, the link generation function can bring very high search benefits and it should be the focus of investments. The link generation function can bring very high search benefits following a complex implementation effort. Whirlpool is currently trying to develop an application that enables the function allowed by the object grouping metadata. They are trying to do this by creating views of the data, but the problem is that if they use the same data in different structures, they have to redo the view.

Granularity levels are also partially used within their repositories, but this is done via ontologies and often returns chaotic results on data relationships. Having reliable

parenthood relationship metadata is better since it prevents data analysts from doing the same job twice, by developing more ontologies.

The company mainly works with ontologies that allow data from different sources to be pooled together. Ontologies are the results of research projects that then are modified and adapted based on the context. Data mapping does not start from the data, but from the related asset from which it is extracted. The data is then associated with its metadata according to the future analysis for which the data is collected. This logic is typical of schema-on-write databases and therefore different from the schema-on-read data lake logic where the data collection has not a precise purpose yet. Once the ontologies are defined, the data are then displayed differently according to the contexts of use. There are therefore applications that aim to "normalise" the data to facilitate data querying and surfing. So, the most important function for the way Whirlpool actually works is Semantic enrichment, to get basic and detailed information about the data, and Indexing, to be able to exploit ontologies by facilitating searching. This massive use of ontologies creates problems with the semantics used for metadata, often increasing misunderstandings and lengthening the time it takes to search for data in the repository. Added to this, the average level of expertise regarding data management in the market is very low, increasing the difficulties in the development of advanced solutions for data analysis.

4.3.2. Alberto Erisimo & Conte Giovanni – SAP

Within MADE, SAP is responsible for data collection and loading into SAP Ana. So, among the many use cases they work with at MADE they are the players responsible for data collection. Alberto and Conte are both specialized in the application of information systems in the manufacturing industry.

As soon as they were presented with the final metadata classification, they immediately appreciated its goodness and completeness, claiming that it can also be useful in databases, especially when there are many of them (such as at MADE). When

approaching a data set for the first time, all the metadata categories allow moving quickly and comfortably through the repository. In a general sense, they are all useful, especially from the point of view of a data analyst. Not all metadata and functions identified will be used for all analyses, but in the long run assuming different analyses, all functions may find application and usefulness.

When analyzing a large amount of data in the productive field the most critical part, beyond the intra-objects information that sometimes is also too much, is represented by inter-objects and global metadata. These can reveal hidden issues, especially the advanced ones that allow you to understand analogies and differences between data. Today, exploiting existing technologies can be automated the generation of many metadata categories and thus facilitate the implementation of functions. For other metadata, the manual association is still very important. For example, if semantic metadata is assigned by a person who knows the meaning of the data, the domain of the data set becomes suddenly clear also to a person who has no knowledge about it. This is possible if, as in SAP, there are no problems with the semantics used for metadata tags, otherwise the manual assignment of metadata can cause problems with the different terminologies used. SAP does not have any data consistency issues, since the semantics to be associated with tags is standardized.

The goodness of intra-object metadata is fundamental to build the other metadata categories and enabling all functions. Anything that can speed up analysis or allow automatic identification of analogies and differences has aroused a lot of interest, such as Similarity links and Difference links. For example, if there is a downtime (which can be a recorded event, and therefore data) this could be due to a type of piece, a type of steel, or a specific machine setting. Having a tool that automatically correlates these events according to similarities can be definitely effective. The Granularity levels function would also resolve some of the problems currently encountered by the company. For example, by defining the relationships between the data, when a pallet is associated with a metadata if the Parenthood relationship is done properly this

information is also associated with all the products contained in the pallet. This actually does not always happen immediately. Semantic enrichment and Indexing are almost taken for granted, as they are less critical at the moment. Usage tracking is more important in some industries, in others less, depending on the presence of regulatory policies that must be respected (whether audits are conducted on the data used in the analyses).

In a data lake, the problem is when it is needed to put together a well-structured database with a set of unstructured data and events. It must be clear how to characterize them since they have to be represented as data points. It is therefore important to know how to move in the system. This is even more important when using several systems from different manufacturers, as at MADE, where the importance of metadata increases exponentially.

4.3.3. Results Validation

These final interviews allowed us to validate what was state in the previous chapters. Not all metadata is equally useful for analysis. This usefulness can vary depending on the organization or the data analyst who interfaces with the data lake. Consequently, this also applies to the usefulness of the different functions enabled by the metadata. This is because there is not an "optimal" way of working, but each data analyst has their own way of working. Nevertheless, respondents appreciated the distinction between essential, useful and advanced metadata. It is almost impossible to approach a repository without essential metadata, especially to understand what can be extracted from the data and what kind of analysis can be carried out on it. Furthermore, the usefulness of using features such as similarity links, an advanced metadata, is also acknowledged since they allow to quickly identify new spaces for future analysis. However, the most critical issue remains the implementation of advanced features within the data lake, as this requires the development and

integration of multiple technologies. This enforces significant investment in research and development to facilitate, accelerate and optimize search.

5 Metadata model selection

5.1. Choosing the metamodel

In this chapter, starting from the most relevant metamodels present in the literature, we check whether or not the functionalities derived from the previous chapters, are covered by the various metamodels. Analyzing the coverage of these metamodels gave us the possibility to identify which is the one best suited to manage a data lake in an Industry 4.0 environment.

5.1.1. Medal

MEDAL, the P. Sawadogo et al. (2019) metamodel, is based on given object knowledge and the division of metadata into three categories: intra-object, inter-object and global metadata. MEDAL adopts a graph organization. An object is represented by a hypercube containing hierarchical sub-dimensions that correspond to the versions and representations of an object. Changes and updates are stored and enabled by oriented edges that connect nodes, covering the concept of **data versioning** and to some extent **data provenance**. The concept of horizontal relationships between data is also considered, both at the content level and at the technical feature level also: edges to model **similarity links** and hyperarcs to translate kinship relationships and object groupings. Moreover, thanks to **logs**, it is also possible to keep track of users' activities. Finally, there are also **indexes or semantic enrichment resources**, in the form of knowledge bases, indexes, or event logs. It also remains to date a theoretical metamodel, in that it has not yet been implemented. However, it must be said that it remains very flexible and suitable for handling structured, unstructured and semi-structured data. We will conclude by saying that Medal includes 6,5/8 functionalities, what is missing is the granularity levels and tracking of the physical data provenance which is a part of the data provenance feature.

5.1.2. Handle

HANDLE is the second metamodel considered proposed by R. Eichler (2021). It will be analyzed since it is claimed to be one of the most generic for data lake management, although as we shall see it turns out to be quite comprehensive. This approach allows for recognition and movement through the various **levels of granularity** of data and for capturing metadata at various granular levels. It then allows the categorization of metadata by **indexing** and enrichment of content information with **semantic enrichment.** HANDLE can also be used to model the same metadata in various ways, depending on the intended use, so, HANDLE supports various features of data lakes such as data lake zones, thus the feature called **polymorphism** is covered. This metamodel allows also the recognition of **similarities** between data objects. Finally, it also allows the **control of access** and activities performed by users. Our evaluation shows that it is easily applicable to metadata management use cases, can reflect the content of existing metadata models, and offers additional metadata management capabilities. In conclusion, HANDLE satisfies 6/8 functionalities, since data versioning and data provenance are completely missing.

5.1.3. Ravat & Zhao

F. Ravat & Y. Zhao (2019) proposed a generic and extensible classification for metadata management based on a fundamental concept of the data lakes architecture: the multi-zones. The raw data zone, to ingest raw data on which no transformation or cleaning is performed. A process zone is used to store raw data while processing them, it is like a buffer. The access zone is the proper storage zone, which ensures accessibility to data. The last one is the governance zone, which ensures data quality, security and life cycle. This metamodel allows linking **similar** data basing on intrinsic characteristics and technical one. In this context of existing works such as automatic detection of relationships between datasets and automatic extraction of data structure, metadata **indexes** and **semantic data**. Also the other functionalities are well represented by this

model, except the granularity levels and the data provenance that are not covered by this one, which makes it 6/8.

5.1.4. Diamantini

This metamodel is not ad-hoc for data lake management, it has been created by C. Diamantini (2016), for multidimensional ontologies with hierarchical nodes and multidimensional points. It has been created for contests in which a large number of business analysts need to share a great amount of knowledge, for this reason, we are considering it as a candidate for being the most suitable metamodel for data lakes in industry 4.0. This is an ontology-based OLAP, meaning that dimensions and facts are enriched by **semantic** information or **tags** that help analysts to better understand the domain of the dataset. Moreover, data units are indirectly linked to data files by **similarity links**. This model is important as considers 3 out of 4 of basic features for the data lake: **link generation**, **granularity levels** and semantic enrichment. It is only missing usage tracking, data provenance and data versioning as functionalities.

5.1.5. GOODS & CoreKG

Both GOODS and CoreKG are black boxes metadata systems since they have very few details about metadata conceptual organization. GOODS is used by Google engineers in managing great amounts of data sets. For this reason, there are not a lot of technical information shared about its functioning. This model extracts metadata ranging from salient information about each dataset (owners, timestamps, schema) to relationships among datasets, such as **similarity** and **provenance**. It is so one of the only metadata models to consider provenance as functionality. It is used so to infer and crawl metadata for billions of datasets, so it is thought properly for data lakes. R. Halevey (2016), quotes that this model has some conceptual holes in polymorphism and granularity levels functionalities. As for CoreKG, A. Behesthi at al. (2018) quotes that it is an open-source solution for managing multiple databases, from relational to noSQL, offering solutions that satisfy **semantic enrichment**, **indexing**, **similarity links**

between different datasets and **data polymorphism**. Unfortunately for those two models, very little technical information about the organization and the structure of the metadata are present.

5.1.6. Ground & GEMMS

Ground, J. Hellerstein at all (2017), and GEMMS, C. Quix at all (2018), are two basic and generic metamodels used for data lake management. While the first is not specifically designed to be used with data lakes, the second one is ad-hoc for this purpose, but Ground appears more extensive than GEMMS since can allow also data versioning and usage tracking. But, both metamodels are considered unsuitable for managing a data lake in an Industry 4.0 environment, as they lack a key feature: similarity links. Neither of these models is able to relate different data objects based on semantic or technical category characteristics. According to the interviews done and addressed in the previous chapters, it has always emerged how important it is to have a cross-sectional view, thus being able to relate data from different silos, finding patterns and similarities useful to create valuable analysis. Similarity links are therefore considered too important a feature to consider a model that does not include it. Consistent with what was expressed by both data scientists and area managers, GEMMS and Ground will not be considered as data models for the final choice.

5.2. The chosen metamodel choice: goldMedal

It is important to say that the models considered are those that we found to be most complete in terms of functionality. Other models were screened but discarded as significantly less complete than those listed above.

Table 19. Metamodels features benchmarking

	Semantic Enrichment	Data indexing	Link generation	Granularity levels	Data provenance	Data polymorphism	Data versioning	Usage tracking
goldMedal	x	x	x	x	y	x	x	x
Medal	x	x	x		y	x	x	x
Handle	x	x	x	x		x		x
Ravat & Zhao	x	x	x			x	x	x
GEMMS	x	x		x				
Ground	x	x					x	x
Diamantini	x	x	x	x		x		
GOODS	x	x	x		x		x	x
CoreKG	x	x	x			x		x
	Basic Functionalities					Advance Functionalities		

Tabel 19 report the features offered by each metadata model. In the table, "X" is assigned to metadata models that can fully deliver a given function, while the "Y" are features partially satisfied as discussed in previous chapters.

As can be seen none of the metamodels satisfies all the 5 basic features. The one that is more complete with basic features is goldMedal proposed by E. Scholly et al. (2021) which has 4,5/5 features satisfied. There is a trade-off between implementing a simpler metamodel without some fundamental basic features and implementing a complete metamodel like goldMedal which has a higher complexity in being implemented. The complexity of metadata model implementation depends on the number of features this can offer. From our side, the importance of having a metamodel with almost all basic features, avoiding turning the lake into a data swamp, is so huge that our choice falls on goldMedal even if may be more complicated than the others. The fact that it also enables "advanced" features may be an advantage when the number of data increases in future or if there is a need to implement advanced features.

From what we see by analyzing compliance with the previously identified requirements, we find that goldMedal is the most complete, detailed and clear metamodel. GoldMedal is the evolution of Medal and some differences are present between the two. In Medal, data items were considered either as raw data or as versions or representations derived from raw data. The concepts of version and representation were used to express updated and transformed data, respectively. But in data lakes, more data items were possible, e.g., temporal representations, or KPIs of different dimensions. Therefore, in goldMedal these concepts are generalized into a global concept named data entity. Some new concepts are added to the newly updated metamodel and the Medal graph logic is changed.

With goldMedal four logical and conceptual concepts are introduced in order to allow data lake management: data entity, grouping, link and process. Those concepts are characterized by attributes or properties that constitute their internal metadata and act together in fulfilling the data lakes' requirements:

- Update and transformation operations that served to track the lineage of representations and versions, respectively, as well as parenthood relationships that express fusion operations, into the concept of process, thus allowing the possibility to satisfy part of provenance as a feature.
- Manages the dynamic of data, thus allowing the possibility to recognize if data has been altered to fulfill requirements of the zone in which it is found. So, data polymorphism is well satisfied.
- Possibility to recognize different granularities of the same object thanks to the Grouping concept. This functionality joined with the possibility to move across the hierarchical scale of a data object, fulfills the requirements of granularity levels as a feature.
- Another change done with goldMedal is considering similarity links into the global concept of link.

So, the only feature that is still unsatisfied, is a part of the data provenance: the tracking of the physical provenance of the data. The other requirement of the data provenance feature is well satisfied since, in goldMedal, the quantitative changes and updates are stored and measured. We believe that goldMedal will be the perfect metamodel for a dynamic and interconnected environment like MADE 4.0 and more in general for industry 4.0 since its logic appears well organized and flexible in dealing with structured, semi-structured and unstructured data. Furthermore, it supports almost all the features explained in chapter 3.3, making it the most generic metadata model to our knowledge. GoldMedal creators state that, in order not to make it become another black box, they will take great care in accompanying users in the appropriation of analysis tools, helping users by intervening when needed with new research. So, it is not just a theoretical model like Medal was. This metadata model further allows non-technical users to access the data lake to ensure the applicability of the metamodel also in an industrialized context, by facilitating the data catalog integration.

6 Conclusion and future developments

The requirements and needs for implementing a data lake change depending on the industry or context in which they are used. Data management within the data lake is still the main challenge for an effective implementation. A poor choice for metadata management can turn the entire data lake into a so-called "data swamp", a repository of data in which it is difficult to find what you are looking for. To avoid this, it is necessary to implement a metadata model enabling easy navigation, search, and discovery of data. Metadata allows more information regarding the meaning, properties, and similarities between data items. All of this has the ultimate goal of facilitating analysis, allowing to gain more knowledge from the data with less effort. The selection of the most appropriate metadata model depends on many factors such as the data used and the analyses that are done.

Through the work done in this dissertation, it has been possible to define the utility of different metadata categories and data lake features useful in Industry 4.0 environments, selecting goldMedal as the most appropriate metadata model in these contexts.

6.1. Metadata categories utility

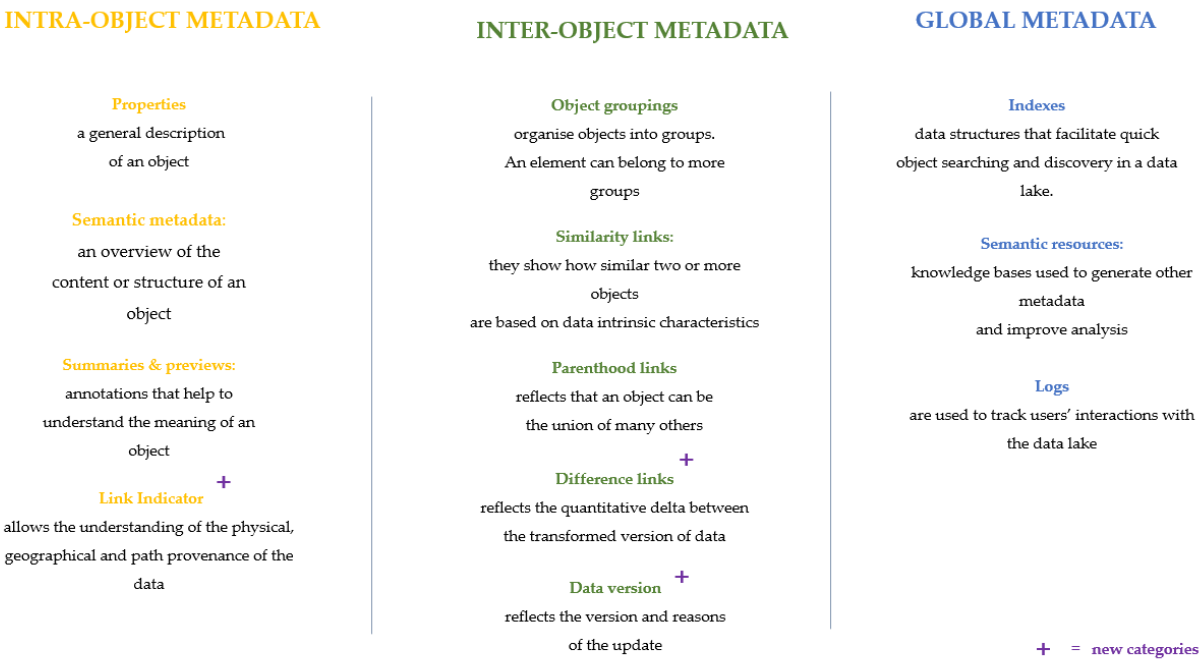
After an extensive study regarding the existing literature about data lakes, we identified the most cited and comprehensive metadata classifications of Sawadogo et al. (2019) and A. Oram (2015). These classifications were the starting point for interviewing area managers of MADE, Politecnico di Milano's I4.0 competence center. The multifunctional nature of MADE as a use case allowed the interviews to be structured for extracting knowledge regarding the use of data in different industrial settings.

Through this information, we were able to define three different levels of metadata utility: essential, useful, and advanced metadata. Essential metadata are those that would make searches impossible if absent, such as properties, and those that make it easier to navigate through the data based on the current way area managers work, such as parenthood relationships. Useful metadata, on the other hand, are that information that is recognized as being highly useful. These provide a better understanding of the meaning of the data, such as semantic metadata and/or facilitate data discovery, like indexes. Last, we have advanced metadata that are all those metadata that can be generated automatically. These go to enrich the set of information associated with a piece of data, leveraging ontologies or identification technologies of patterns, images, text... These metadata, while very useful and will be increasingly used in the future, to date in I4.0 contexts are considered advanced information.

Before cross-referencing the metadata with their enabling features, it was necessary to enrich Sawadogo et al.'s classification with three more metadata. Following the interviews, we noticed that the classification used lacked information regarding data versions, provenance, and differences between data. This led us to the identification of three new metadata categories: Data version, Link indicator, and Difference links. Data version gives information about the version and the reasons for an update, Link indicator allows to understand the path provenance of data, and Difference links, reflect quantitative information about the delta between different data versions.

Once this was done and after harmonizing the new metadata categories with Sawadogo et al.'s classification (making the new metadata categories fall into intra-object, inter-object, and global metadata) we defined our metadata set as shown in Figure 4.

Figure 4. Final metadata categories set integrated with Sawadogo et al. classification



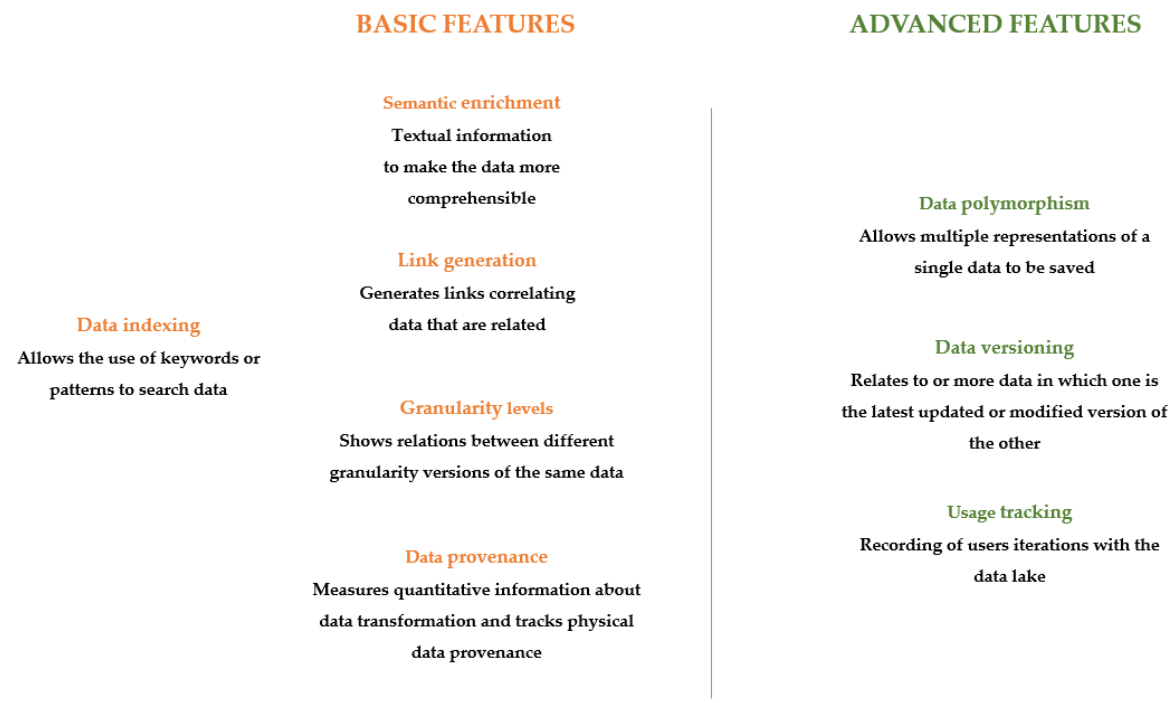
6.2. Metadata features utility

Once we had defined all the metadata categories and assigned the utility resulting from the interviews to each of them, we had to figure out what features a metadata model must have in I4.0 industries. The operation of these features is strictly dependent on the metadata that enables their function. This means that metadata serves as input for the proper functioning of a given feature. To identify all the features that a metadata model can offer we intersected the work done by Sawadogo et al. (2019) and R. Eichler et al. (2020). They both proposed two different classifications of metadata model features. Looking at the similarities and differences, we identified a set comprising eight features: Semantic Enrichment, Data indexing, Link generation, Data polymorphism, Data versioning, Usage tracking, Granularity levels, and Data provenance.

To understand which features result to be most useful according to our use case, it was necessary to translate the usefulness of metadata into the usefulness of features. To do this, we held brainstorming sessions to identify which metadata are enablers of a

particular function. One metadata can enable multiple functions, and likewise, one function can be enabled, partially or fully by multiple metadata. To turn metadata ratings on the features, it was decided to select the highest exclusive enabling metadata category rating. This method allowed ratings to remain aligned with the judgments observed during the interviews of area managers. This allowed us to identify what features are needed in I4.0 environments thus understanding their usefulness. More specifically, we can distinguish between basic features and advanced features. Basic features are all those functions that would make the data lake unusable if absent, turning it into a data swamp. Advanced features, on the other hand, facilitate the search and organization of data when its number increases considerably, so also as the number of unstructured data stored increases. Figure 5 summarizes our classification for data lake features in I4.0 contexts.

Figure 5. Basic and advanced features in I4.0 environments



6.3. Metadata model selection

All this led to the identification of a set of metadata models for data lake management. Thanks to the information regarding the usefulness of the features, it was then possible

to define an "optimal" metadata model for our case. The metamodels were compared, checking if the functionalities identified in the previous chapter were satisfied by the model or not. The criterion of the selection was to prefer metamodels with the most basic functionalities satisfied. Despite this, it has been noticed that no metamodel had all the five basic required functionalities of a data lake in I4.0 contexts. The most complete metamodel, goldMedal, has 4,5 out of 5 basic functionalities satisfied since provenance is not fully ensured. This is because the tracking of the physical provenance of data is not present. Even if the model had 3/3 advanced functionalities that may result difficult to implement, it has been selected since a lot of work has been done by goldMedal creators to ensure non-technical users access the data lake. In conclusion, goldMedal has been selected as the optimal metamodel to manage data in an industry 4.0 environment.

6.4. Study implication

6.4.1. The starting point for data lake implementation in I4.0 organizations

A data lake without effective metadata management risks becoming unusable. The selection of the logical framework for effective metadata management depends on the industry in which the organization operates. As a result of the work done so far, the fundamentals for designing a data lake in I4.0 and IoT-related fields have been laid. It is to be considered as a starting point for the applicative development of a data lake. Once the metadata model is clear, it will then be necessary to build and program the data lake considering the metadata organization identified. Doing so will require having appropriate skills and selecting the right tools to build the final solution.

Work discussed in the chapters will allow greater productivity from the use and analysis done with the data lake. Analysts will be able to quickly discover the data they are searching for, speeding up analysis and decreasing the workforce needed.

Valuable data will not be left unused and human effort spent on non-value-added activities will be decreased.

6.4.2. Metadata categories in databases

The analyses regarding the use of different metadata categories can also be useful when applied to traditional data warehouses. The metadata categories seen allow for the organization and ease of searching for data also within DWs. The only difference is that usually with DWs the data are saved already knowing the purpose for which they are collected, thus reducing the usefulness of a precise metadata model for organizing the data. Despite this, data can also be reused for future research. In this case, as the information about the data increases, it will be more easily understood, found, and thus reusable.

6.4.3. Data value in industrial contexts

To date, thanks to new methodologies for extracting insights from data, they are becoming increasingly valuable to companies. The industrial value of having a data repository in which you cannot find what you are looking for is zero. Some analyses may also not be feasible given the inability to find the data of interest. Due to the work done this risk decreases by selecting the most tailored metadata model for the context of use. This will optimize and speed up data searching. By taking advantage of automatic metadata generation functions (like similarity links, difference links, and semantic metadata...) it will also be possible to identify clusters of data and take cues for future analysis more easily. Thus, a well-structured and organized data lake enables to increase the value associated with data. This is true even if the company shares or sells data with third parties. These will be able to derive value and better understand the content of the data set, also optimizing their analyses.

6.4.4. FAIR principle enforcement

A well-designed data lake also enforces the four FAIR principles (Findability, Accessibility, Interoperability, Reusability). Thanks to the choice of the right metadata model all the points are reinforced. Findability, which indicates the findability of the data, is greatly increased since each data item is assigned all the information needed to track the data in the system. Accessibility is guaranteed to all users who are entitled to access a given data set. Interoperability, which indicates the integration between different data is enhanced by functions such as Link generation that allows data to be correlated. Usability is ensured since the data and metadata are well described so that they can be reproduced and combined in different contexts.

6.5. Study limitations

6.5.1. Study limited to a single use case

Our study is based on information obtained from the literature and interviews with MADE area managers. MADE represents an excellent use case for obtaining information on data analysis in industrial fields given its multipurpose nature divided into six areas. Despite this, increasing the number of actors interviewed would allow to obtain additional information on data analysis outside this organization. This could uncover different perspectives since, as discovered during interviews with companies, the approach to data management changes greatly between companies.

6.5.2. A theoretical work

Although the literature and knowledge regarding data lakes are constantly evolving there are few real-world examples of metadata model implementations. This is to be expected as data lake technology is relatively new and the literature is constantly evolving. Another problem is that many data lake producers do not share information about their metadata model and what metadata is used in their systems. This does not

allow for information regarding metadata management used by the world's top players. Since this is a theoretical work all the analyses done on metadata, features and metadata model are perfectly implementable on most open-source libraries (such as Apache Hadoop). How to do this remains an open issue and it must be a subject of analysis in future research.

6.5.3. Apache Hadoop plug-ins identification and implementation

The implementation of most of the proposed features requires the integration of several additional modules into Apache Hadoop. Link generation function, for example, is still missing in companies like Whirlpool and this must be the field where they have to invest. With different levels of complexity, it is needed to work to be able to develop the technology needed to implement the different functions.

The same applies to the automatic assignment of certain data categories. For advanced metadata to work properly, for example, ontologies must be developed in order to implement semantic metadata and indexes. For metadata such as Similarity links, on the other hand, AI technologies need to be developed and integrated to be able to recognize data with similarities.

Another open issue remains the integration of the metadata system with the data catalog. This tool allows non-data analyst users to easily navigate the data lake, making the system easier to navigate and increasing the number of users who can interact with it. The goodness of the data catalog strictly depends on how well the metadata model used is integrated with it.

Bibliography

L. Zhou, W. Tu, C. Wang, and Q. Li, "A Heterogeneous Access Metamodel for Efficient IoT Remote Sensing Observation Management: Taking Precision Agriculture as an Example," vol. 9, no. 11, pp. 8616–8632, Jun. 2022, doi: 10.1109/JIOT.2021.3118024.

C. Giebler, C. Gröger, E. Hoos, H. Schwarz, and B. Mitschang, "Leveraging the Data Lake: Current State and Challenges." Springer International Publishing, p. 179, 2019, doi: 10.1007/978-3-030-27520-4_13.

P. N. Sawadogo, É. Scholly, C. Favre, É. Ferey, S. Loudcher, and J. Darmont, "Metadata Systems for Data Lakes: Models and Features." Springer International Publishing, p. 440, 2019, doi: 10.1007/978-3-030-30278-8_43.

R. Eichler, C. Giebler, C. Gröger, H. Schwarz, and B. Mitschang, Big Data Analytics and Knowledge Discovery : 22nd International Conference, DaWaK 2020, Bratislava, Slovakia, September 14–17, 2020, Proceedings, vol. 12393. Cham: Springer International Publishing, 2020.

F. Ravat and Y. Zhao, "Metadata Management for Data Lakes." Springer International Publishing, p. 37, Sep. 01, 2019, doi: 10.1007/978-3-030-30278-8_5.

P. Sawadogo and J. Darmont, "On data lake architectures and metadata management," vol. 56, no. 1, pp. 97–120, 2021, doi: 10.1007/s10844-020-00608-7.

S. Gidley, "Tips for managing metadata in a data lake" 2017.

B. Pernici, "Message from the Guest Editor," vol. 13, no. 2, p. 12, Aug. 1992, doi: 10.1145/134376.1017801.

Huang Fang, "Managing data lakes in big data era: What's a data lake and why has it became popular in data management ecosystem," Jun. 2015, pp. 820–824, doi: 10.1109/CYBER.2015.7288049.

M. Armbrust, A. Ghodsi, R. Xin, and M. Zaharia, A New Normal?, 1st ed. Princeton University Press, 2019, p. 206.

C. Opatija, “2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO 2018) Pages 1-805.” .

S. Jaskó, A. Skrop, T. Holczinger, T. Chován, and J. Abonyi, “Development of manufacturing execution systems in accordance with Industry 4.0 requirements: A review of standard- and ontology-based methodologies and tools,” vol. 123, p. 103300, Dec. 2020, doi: 10.1016/j.compind.2020.103300.

“Industry 4.0 How to navigate digitization of the manufacturing sector.” .

H. Dibowski and K. Kabitzsch, “Semantic device descriptions based on standard Semantic Web technologies,” May 2008, pp. 395–404, doi: 10.1109/WFCS.2008.4638720.

J. . Hellerstein, “Achieving service rate objectives with decay usage scheduling,” vol. 19, no. 8, pp. 813–825, Aug. 1993, doi: 10.1109/32.238584.

“Awel Eshetu Fentaw Data Vault Modelling An Introductory Guide.” Mar. 01, 2014.

H. Zaidi, Quantification of Small-Animal Imaging Data. New York, NY: Springer New York, 2014, pp. 467–494.

M. Chen, S. Mao, Y. Zhang, and V. C. M. Leung, “Challenges and Future Prospects.” .

I. Grangel-González et al., “Semantic Data Integration for Industry 4.0 Standards.” Springer International Publishing, p. 230, 2017, doi: 10.1007/978-3-319-58694-6_36.

W. Scheidel, “Roman Real Wages in Context.” SSRN, 2010, doi: 10.2139/ssrn.1663559.

N. Miloslavskaya and A. Tolstoy, “Big Data, Fast Data and Data Lake Concepts,” vol. 88, pp. 300–305, 2016, doi: 10.1016/j.procs.2016.07.439.

R. Eichler, C. Giebler, C. Gröger, H. Schwarz, and B. Mitschang, “Modeling metadata in data lakes—A generic model,” vol. 136, p. 101931, Nov. 2021, doi: 10.1016/j.datak.2021.101931.

R.-M. Holom, K. Rafetseder, S. Kritzinger, and H. Sehrschön, “Metadata management in a big data infrastructure,” vol. 42, pp. 375–382, 2020, doi: 10.1016/j.promfg.2020.02.060.

M. Marjani et al., “Big IoT Data Analytics: Architecture, Opportunities, and Open Research Challenges,” vol. 5, pp. 5247–5261, 2017, doi: 10.1109/ACCESS.2017.2689040.

B. Pernici, “Message from the Guest Editor,” vol. 13, no. 2, p. 12, Aug. 1992, doi: 10.1145/134376.1017801.

“MADE Aree e dimostratori.” – Politecnico di Milano - MADE

C. Cappiello, A. Gal, M. Jarke, J. Rehof, J. Rehof, and T. U. Dortmund, “Data Ecosystems: Sovereign Data Exchange Among Organizations (Dagstuhl Seminar 19391),” doi: 10.4230/DagRep.9.9.66.

C. Quix, R. Hai, and I. Vatov, “Metadata Extraction and Management in Data LakesWith GEMMS,” no. 9, pp. 67–83, Dec. 2016, doi: 10.7250/csimq.2016-9.04.

I. Nogueira, M. Romdhane, and J. Darmont, “Modeling Data Lake Metadata with a Data Vault,” Jun. 2018, pp. 253–261, doi: 10.1145/3216122.3216130.

K. Lock and H. Ellis, The New Towns Today, 1st ed. RIBA Publishing, 2020, pp. 59–78.

B. Laplante and B.-P. Hebert, “An Introduction to the Use of Linear Models with Correlated Data,” vol. 28, no. 2, pp. 287–311, Dec. 2001, doi: 10.25336/P6CC87.

"Ontology 2 : the Real Semantics Book Data Lakes, Data Ponds, and Data Droplets The inevitability of Data Lakes Technical trends leading to Data Lakes: Hardware." .

R. Eichler, C. Giebler, C. Gröger, H. Schwarz, and B. Mitschang, Big Data Analytics and Knowledge Discovery : 22nd International Conference, DaWaK 2020, Bratislava, Slovakia, September 14–17, 2020, Proceedings, vol. 12393. Cham: Springer International Publishing, 2020.

T. Burns, J. Cosgrove, and F. Doyle, "A Review of Interoperability Standards for Industry 4.0," vol. 38, pp. 646–653, 2019, doi: 10.1016/j.promfg.2020.01.083.

P. Sawadogo and J. Darmont, "On data lake architectures and metadata management," vol. 56, no. 1, pp. 97–120, 2021, doi: 10.1007/s10844-020-00608-7.

C. Diamantini, P. L. Giudice, L. Musarella, D. Potena, E. Storti, and D. Ursino, "A New Metadata Model to Uniformly Handle Heterogeneous Data Lake Sources." Springer International Publishing, p. 165, Oct. 12, 2018, doi: 10.1007/978-3-030-00063-9_17.

E. Scholly, C. Favre, E. Ferey, and S. Loudcher, "HOUDAL : A Data Lake Implemented for Public Housing," 2021, doi: 10.5220/0010418200390050.

Y. Park, J. Choi, and J. Choi, "Conceptual metadata model for sensor data abstraction in IoT environments," vol. 383, no. 1, p. 12013, Jul. 2018, doi: 10.1088/1757-899X/383/1/012013.

H. J. Lee and M. Sohn, "Construction of Tag-Based Dynamic Data Catalog (TaDDCat) Using Ontology." IEEE, Sep. 01, 2012, doi: 10.1109/nbis.2012.116.

M. Khan, Xiaotong Wu, Xiaolong Xu, and Wanchun Dou, "Big data challenges and opportunities in the hype of Industry 4.0," May 2017, pp. 1–6, doi: 10.1109/ICC.2017.7996801.

P. Sawadogo and J. Darmont, "On data lake architectures and metadata management," vol. 56, no. 1, pp. 97–120, 2021, doi: 10.1007/s10844-020-00608-7.

List of Figures

List of Tables