DERM: A Reference Model for Data Engineering

Daniel Tebernum[1], Marcel Altendeitering[1] and Falk Howar[2]
[1] *Data Business, Fraunhofer ISST, Emil-Figge-Strasse 91, 44227 Dortmund, Germany*
[2] *Chair for Software Engineering, TU Dortmund University, Otto-Hahn-Strasse 12, 44227 Dortmund, Germany*

Keywords: Reference Model, SLR, Data Lifecycle, Data Engineering, Research Map.

Abstract: Data forms an essential organizational asset and is a potential source for competitive advantages. To exploit these advantages, the engineering of data-intensive applications is becoming increasingly important. Yet, the professional development of such applications is still in its infancy and a practical engineering approach is necessary to reach the next maturity level. Therefore, resources and frameworks that bridge the gaps between theory and practice are required. In this study, we developed a data engineering reference model (DERM), which outlines the important building-blocks for handling data along the data lifecycle. For the creation of the model, we conducted a systematic literature review on data lifecycles to find commonalities between these models and derive an abstract meta-model. We successfully validated our model by matching it with established data engineering topics. Using the model derived six research gaps that need further attention for establishing a practically-grounded engineering process. Our model will furthermore contribute to a more profound development process within organizations and create a common ground for communication.

1 INTRODUCTION

The ability to efficiently utilize information and knowledge for competitive advantages is vital for organizations and forms an important organizational capability (Levitin and Redman, 1998). Data is the foundation for information and knowledge and must therefore be managed appropriately to support organizational decision making and success (Fisher, 2009). For treating data as an asset, several methods and frameworks have emerged in the information systems and business related research fields. These approaches often focus on specific data types (e.g. master data (Otto, 2015)) and put managerial measures (e.g. information governance (Tallon et al., 2013)) at the forefront. Adequate actions are therefore primarily dedicated to the managerial level within organizations (Khatri and Brown, 2010; Amadori et al., 2020).

However, with the prevalence of data-intensive applications (e.g. machine learning or IoT applications), there is a need to create awareness of adequately handling and managing data not only at a managerial level, but also for software engineers (Kleppmann, 2017; Amadori et al., 2020). At the same time, data engineers are primarily concerned with preparing data for data scientists but neglect important software engineering practices (Kleppmann, 2017). We therefore argue that the bridge between the data engineering and software engineering communities needs to be crossed with a critical rethinking of the currently established engineering of data-intensive applications. For this purpose, a common ground for a practical data engineering process needs to be established that takes the full data lifecycle into account. To the best of our knowledge, such a practical engineering process does not yet exist in literature. Specifically, it is necessary for two reasons. First, data processing is becoming increasingly important in application development and software engineers need to operate with novel data structures and volumes (e.g. big data, data streams) (Chen et al., 2013). To do so, they need a better understanding of designing data infrastructures and systems. Second, the establishment of a professionalized development process similar to the software engineering discipline is necessary. Therefore, resources that bridge the gaps between theory and practice are required to overcome the “one size fits all” approach that is currently in place (Stonebraker and Çetintemel, 2018). This way, the data engineering process can help to overcome typical real-world problems and operationalize the software creation process in the light of new data sources, such as big data.

The following example summarizes our research

Tebernum, D., Altendeitering, M. and Howar, F.
DERM: A Reference Model for Data Engineering.
DOI: 10.5220/0010517301650175
In *Proceedings of the 10th International Conference on Data Science, Technology and Applications (DATA 2021)*, pages 165-175
ISBN: 978-989-758-521-0

motivation and intention: In the machine learning domain, the success and usefulness of prediction models is currently measured using accuracy measures, such as the F-Score. Other important aspects, like how maintainable it is, how it needs to be secured, or where it is stored, are often disregarded. Considering such aspects is nevertheless important to move away from individual data science projects towards an engineering discipline for machine learning models. This way, machine learning can be successful and efficient at a larger scale. The same aspects do not only apply to machine learning, but should be raised during the creation of any data-intensive software artifact.

With our research, we aim to contribute to the establishment of a data engineering reference model. We argue that such a model must adhere to the data lifecycle and provide answers to the questions raised in the different phases. We thus conducted a Systematic Literature Review (SLR) to analyze the current state of research on data lifecycles and formulate an abstract data lifecycle. This review provides us with the necessary information about what aspects need to be adressed in developing data-intensive applications. Furthermore, we are formulating concrete themes and name aspects that software engineers should take into consideration when developing data-intensive applications. Specifically, we aim to answer the following research questions:

- RQ1: *What are the building blocks of a data engineering reference model?*
- RQ2: *Can a data lifecycle be used as a foundation for a data engineering reference model?*
- RQ3: *Can we use the reference model to identify possible research gaps?*

The remainder of this paper is structured as follows. We start with a description of our research methodology that we followed in our study in Section 2. In Section 3, we outline how our reference model was developed and go into details about the elements contained. We then validate the model in Section 4 and derive open research gaps in Section 5. We finish our paper with a conclusion in Section 6, also addressing the limitations and possible future work.

2 RESEARCH METHODOLOGY

The development of a reference model for data engineering and finding an answer to RQ1 requires an understanding of what distinctive data challenges are raised during development. We thus decided to focus our research on the review of existing data lifecycles and develop a generic data lifecycle. Based on this lifecycle, we then formulate what challenges and aspects should be incorporated in data engineering projects and how it can contribute to the successful engineering of data-intensive applications.

For reviewing existing data lifecycles in literature, we conducted a SLR as described by (Kitchenham, 2004) and (Kuhrmann et al., 2017). According to (Kuhrmann et al., 2017), a SLR is well suited for identifying, analyzing, and interpreting existing knowledge in an unbiased and reapeatable way (Kuhrmann et al., 2017). We separated the review process into the three distintive steps: *plan*, *execution*, and *review* as recommended by (Kitchenham, 2004).

2.1 Plan

We initiated our research process by selecting appropriate data sources. Therefore, we adapted the most common sources in the computer science domain as defined by (Kuhrmann et al., 2017). This selection yielded in the following seven databases: IEEE Xplore, ACM Digital Library, Science Direct, SpringerLink, Wiley, DBLP, and Scopus. In the next step, we defined the following search terms as relevant to our study: *lifecycle* and *data*. Following the guidelines of (Kuhrmann et al., 2017), we initially tested different search queries to find one that is suitable for our research. For this, we used the search engine Scopus and entered different combinations of *lifecycle* and *data*. We observed that using the word *data* as a single term produces a very large result set. Consequently, we limited our search to direct combinations of the search terms and formulated the following search expression: ("datalifecylce" OR "data lifecycle" OR "data-lifecycle" OR "data-life-cycle" OR "data life cycle"). We then applied the search expression to the different query languages of the selected databases.

2.2 Execution

Using the aforementioned search expression, our initial search resulted in 515 articles across all databases (Step 1). In this step, we ensured that the papers were written in English and that the full-text was available. Afterwards, we limited our results to conference and journal papers and excluded duplicates from the set of papers (Step 2). This step reduced the number of papers to 359. The manual paper selection process began by reviewing the papers based on their titles and abstracts, which resulted in 57 papers (Step 3). In Step 4, we manually reviewed and voted on the remaining 57 paper, which further reduced the number of papers to 27. In this step, we used a majority voting principle including the three authors (Kuhrmann et al., 2017). Hereby, the first two authors voted individually on each paper and the third author voted on papers that were still undecided. Following the guidelines of (Webster and Watson, 2002), we also conducted a forward and backward search on the 27 identified paper to include papers relevant to our study. This step led to another seven papers we identified and results in a total of 34 papers included in our literature review. Figure 1 summarizes our paper selection process.

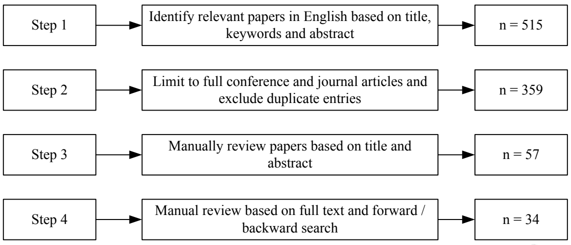
Step 1
Identify relevant papers in English based on title, keywords and abstract
n = 515
Step 2
Limit to full conference and journal articles and exclude duplicate entries
n = 359
Step 3
Manually review papers based on title and abstract
n = 57
Step 4
Manual review based on full text and forward / backward search
n = 34

Figure 1: Paper selection process.

There were several reasons for papers to be excluded in Step 3 or 4. The most common were: The presented lifecycle is too specific or focused on certain aspects (e.g. security lifecycles or lifecycles in Biology). The paper describes an architecture or software rather than the data lifecycle itself. A different lifecycle (e.g. product lifecycle) was described and data was only an aspect within this lifecycle.

2.3 Review

For data synthesis and the subsequent analysis and reporting of our research findings, we used open coding of Grounded Theory Methodology (GTM) as an example (Strauss and Corbin, 1997). Specifically, we wanted to categorize the phases and elements of data lifecycles that are discussed in literature into different abstract themes. We initially generated codes based on the actions performed on data, such as reading a file, removing data errors, vizualising numbers, or sharing data sets with colleagues. However, we observed that the papers not only describe the actions performed on data but also specify the context, which affects the data, like organizational guidelines or certain technologies. There was no common discriminator for the contexts so we adopted contexts that are frequently mentioned in the papers. The difference between the two categories is that a context can apply to several actions. For example, a data management tool could support multiple actions (e.g. access, use, and destroy) performed on data. We thus decided to summarize the contexts that are used as layers in the data lifecycle and code the papers for both, the actions that are performed on data and the layers that are used.

We then continued by reducing these descriptive codes to interpretative clusters that form abstract themes of organizational actions and layers (i.e. the contexts) on which these are happening (Miles and Huberman, 1994). The identified themes were generic in the sense that they occurred in multiple papers. This led us to the conclusion that they play an important role in the general data lifecycle and should be considered in the engineering of data-intensive applications. To check whether our themes were internally consistent and our derived themes were discrete, we constantly asked ourselves the questions: “Is this code similar to that code?” and “Are these codes different from those codes?” as described by (Jarzabkowski, 2008). Sparse codes (e.g. describing technical details) that did not match any cluster were discarded as they were either very specific or did not match our research objective.

We initiated the coding process by applying the coding scheme to a small subset of five papers. This was done to gain a better understanding of the data lifecycles and test our coding scheme on papers from our literature review. After the initial coding, the first two authors independently coded the relevant aspects in the remaining papers in different abstract themes. Potential conflicts during the coding and naming of the identified themes were clarified due to discussions among the researchers until a full consensus was reached.

Following the descibed coding procedure, we identified six abstract themes for actions performed on data, namely: *Plan, Create, Access, Use, Transform,* and *Destroy*; and four abstract themes for the layers on which data is handled: *Metadata, Technology, Data Quality,* and *Enterprise*. Table 1 maps the papers to their respective themes. The way we used our results for building a reference model and a detailed description of the themes and subcategories is available in Section 3.

3 DERM: DATA ENGINEERING REFERENCE MODEL

To the best of our knowledge, the data engineering community lacks a reference model that can be utilized as a common ground for the engineering of data-intensive applications. We argue that the data lifecycle is the core element of such a reference model and can be used to assign the currently established research and working topics. Based on our results in

Table 1: Overview of data lifecycle literature.

	Lifecycle Phases						*Lifecycle Layers*				
Paper	**Plan**	**Create**	**Select/Access**	**Use**	**Transform**	**Destroy**	**Metadata**	**Data**	**Technology**	**Enterprise**	**Special focus**
(Tripathi and Pandy, 2018)	x		x	x	x		x	x			Research
(Elsayed et al., 2008)	x		x	x				x		x	Research
(Huang et al., 2019)	x		x	x	x			x		x	Big Data
(W3C, 2014)	x	x	x	x			x	x		x	Linked Data
(Yazdi, 2019)	x		x	x	x		x	x		x	Research
(Emam et al., 2019)	x		x	x	x		x	x		x	Biology
(El Arass et al., 2020)	x		x	x	x	x		x		x	Big Data
(El Arass et al., 2017)	x	x	x	x	x	x		x		x	
(Sinaeepourfard et al., 2016)	x		x	x	x			x	x	x	Smart City
(Maindze et al., 2019)	x		x	x	x	x	x	x	x	x	
(Möller, 2013)	x	x	x	x	x	x	x	x	x	x	Semantic Web
(DAMA, 2017)	x	x	x	x	x	x		x		x	
(Wing, 2019)		x	x	x				x			Big Data
(Hubert Ofner et al., 2013)	x	x		x			x	x			Master Data
(Xianglan, 2017)			x	x	x		x	x			Coal Mining
(Bychkov et al., 2018)			x	x	x			x	x		Astrophysics
(Solanas et al., 2017)	x		x	x				x	x		Healthcare
(Alladi and Prasad, 2018)			x	x	x			x	x		Big Data
(Pääkkönen and Pakkala, 2015)			x	x	x			x	x		Big Data
(Alshboul et al., 2015)			x	x	x			x	x		Data Security
(Rahul and Banyal, 2020)		x	x	x	x	x		x			Data Analytics
(Polyzotis et al., 2018)			x	x	x			x	x		Machine Learning
(Moiso et al., 2012)			x	x				x	x		Personal Data
(Christopherson et al., 2020)		x	x	x	x	x		x	x		
(Cao et al., 2019)			x	x	x		x	x	x		Data Analytics
(Levitin and Redman, 1993)			x	x			x	x			
(Yu and Wen, 2010)		x	x	x	x	x		x			
(Allanic et al., 2017)			x	x	x			x			Biology
(Grunzke et al., 2015)		x	x	x				x			Natural Sciences
(Liu et al., 2013)		x		x	x			x			Software
(Morris, 2018)		x		x	x	x		x			Biology
(Cheng et al., 2013)		x		x		x		x			Big Data
(Ho and Abramson, 2007)		x			x	x		x			Research
(Simonet et al., 2013)		x	x			x		x			

Section 2, such a reference model should consist of distinctive phases and layers: Phases describe certain actions that are performed on data objects, while layers specify the contexts that affect the phases.

During our SLR, we observed that many papers are domain-specific in their connotations (see special focus in Table 1). As a result, they often contain specific elements that are not needed for a general understanding in the data engineering community or are potentially misleading. For example, (Bychkov et al., 2018) describe an *Education* phase. Obviously, this activity is useful in many areas. However, it cannot be taken as a basic building block because it is specifically dependent on the presence of human actors.

Another reason is the external perspective that these papers adopt. They see the data lifecycle as a part of a software system or overall process that cannot live on its own. This can lead to assumptions and consequently design decisions that are not a necessity but are injected from the environment. For example, (Yu and Wen, 2010; Sinaeepourfard et al., 2016; El Arass et al., 2017) have inserted an *Archive* phase.

While it is not domain specific and makes sense in many cases, it is just a human-made construct to address inherent deficiencies in the surrounding system.

To address these shortcomings and establish a common ground in data engineering, we propose our model **DERM** (Data Engineering Reference Model). The composition and visual appearance of the model was developed in an iterative approach. From the selected papers, we randomly chose one model as our starting point. Ongoing from this, we integrated one model after the other to reach an abstract meta-model. In this process, we used several strategies to achieve our final model (see Figure 2). Our main philosophy was to look at the topic from a data perspective. We continuously asked ourselves “Is the data witnessing this phase?”, “Does this phase make a difference to the data or is it more of a semantic difference?”, “Is this topic really influencing the data?”, and so forth. Then, we checked whether elements were already present or used synonymously. We added elements that were new and determined where to place them due to discussion. Sometimes, elements got merged or removed if they did not fit to our philosophy. We distinguished elements into data related activities (*Phases*) and subject areas (*Layers*). We modeled activities as boxes on a cycle. Over the iterations, we changed the position of the boxes based on the suggestions of the data lifecycles seen so far. We modeled subject areas similar to Venn diagrams. Their positions and intersections changed during the iterations. We tried to fit the phases and layers together in an overlapping way in one model. After the last iteration, we adjusted the visualization of our model for better accessibility.

3.1 Phases

Every single data object passes through several phases that describe what happens to the data object at that point. For the engineering of data-intensive applications, it is vital to be aware of these phases and implement measures dedicated to adequately manage the data in the respective phase. Although the phases are modeled around a cycle and thus follow a path, phases may be skipped if necessary.

Plan. The *Plan* phase comprises several activities that are conducted before the data lifecycle starts. It therefore sets certain guidelines and boundaries that are relevant throughout the data engineering process before the inital creation of data. It is not part of the iterative data lifecycle, but rather a mediating factor. The concrete activities differ based on the context and use case. For instance, (Sinaeepourfard et al., 2016) describe business requirements and scientific demands for this phase, which can influence whether a data source is valid or what data transformations are required. (Tripathi and Pandy, 2018) and (DAMA, 2017) generalize this step as a data management plan that, among others, includes organizational policies, copyright and licensing guidelines, and requirements for documentation.

Create. In the *Creation* phase, new data is being created from scratch, either manually (e.g. (Yu and Wen, 2010)), due to the automatic capture of signals (e.g. (Christopherson et al., 2020)), or the transformation of a previously existing data object. Based on the context the data is used in, the *Create* step can invoke additional subsequent steps. For example, (Cheng et al., 2013) describe data quality and semantical steps like allocation of semantic concepts or association mapping. (Yazdi, 2019) argues that the creation of a data object should be linked with the creation of respective metadata objects to improve the overall data quality.

Select/Access. Generally, the *Select/Access* phase describes the manual or automated identification and access to data objects that are located within or outside the organization. (Alshboul et al., 2015) and (Liu et al., 2013) specify the *Access* phase as the search and acquisition of data and its integration into an organizational database systems. (El Arass et al., 2020) put an emphasis on the user for this phase and describe the need for a suitable interface to the data consumer. They also mention access control rules for data security and usage control depending on the role and rights of the respective user. Another topic that is frequently mentioned in this phase is “data provenance” (e.g. (Allanic et al., 2017; Ho and Abramson, 2007)), which describes the origin of data and when and how it was accessed and changed. Depending on the respective provenance, a data object can be more or less useful for data access.

Use. The *Use* phase comprises all activities that are performed on data. This usually involves data analysis, integration, and visualization steps (e.g. (Polyzotis et al., 2018; Bychkov et al., 2018)). The usage step receives the most attention in the data lifecycle as it can generate novel insights and value for an organization. Hereby, the presentation of the results to the user is vital for the success of the *Use* phase and should be appropriately designed (Levitin and Redman, 1993). It therefore receives much attention, especially in light of the new opportunities of machine learning and artifical intelligence. An

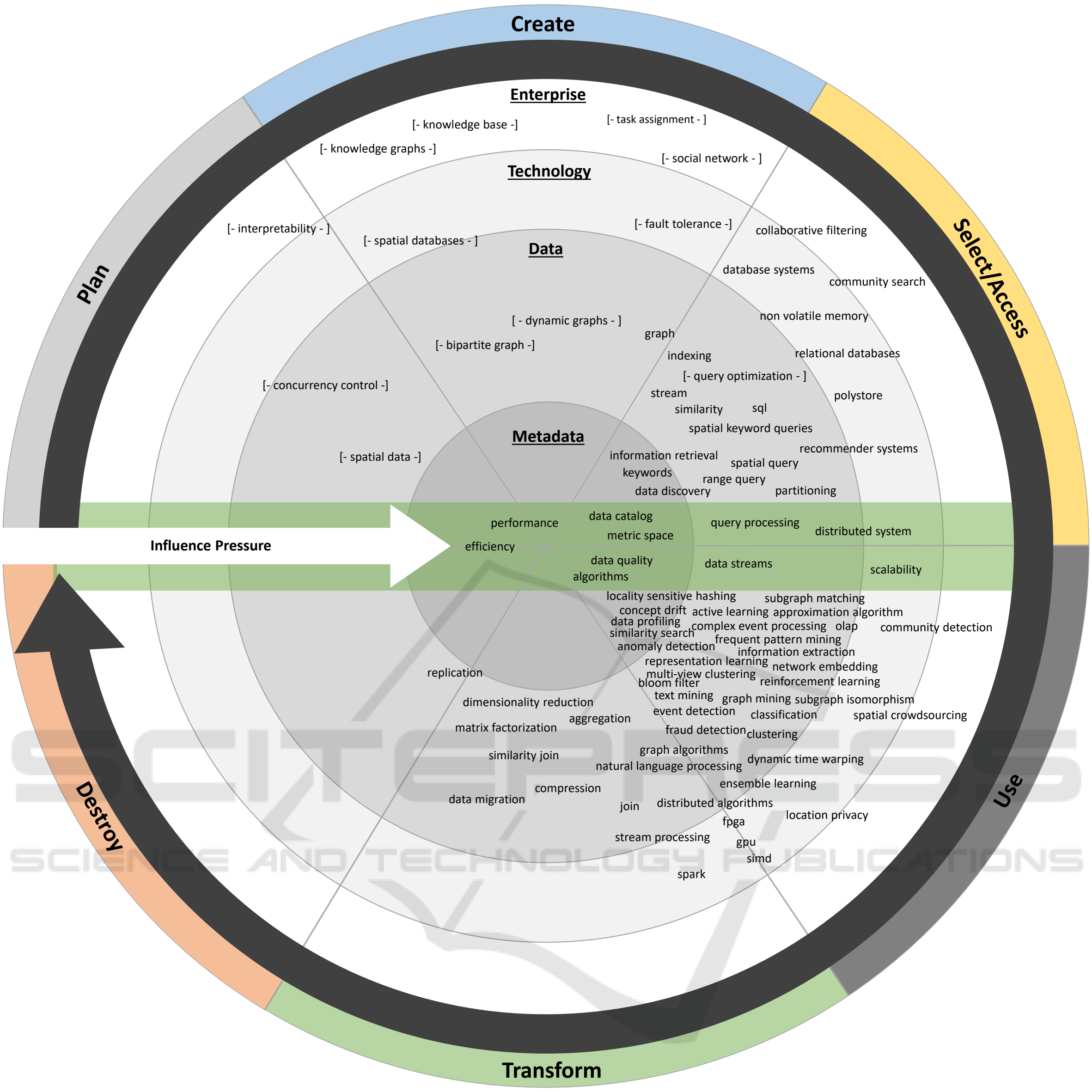

Figure 2: DERM - A Reference Model for Data Engineering.

integral part of data usage is ensuring that the data is of high quality to improve the value of the derived insights. This can, for instance, be achieved with data quality checks or rules (Sinaeepourfard et al., 2016; Hubert Ofner et al., 2013).

Transform. Whenever a data object is changed or updated, the *Transform* stage is invoked. It contains the transformation of the data, which consequently results in the creation of a new data object that is different to the previous one. The *Transform* step can be triggered by different use cases, like data cleaning, formatting, conversion, or standardization (Emam et al., 2019; Christopherson et al., 2020). This step also includes all activities necessary for the long-term storage and preservation of data (Xianglan, 2017).

Destroy. At the end of its lifecycle, data is deleted to provide space for new data objects. However, it is difficult to determine the moment a data object should be deleted. For example, (El Arass et al., 2020) define the point of destruction as the one where data has become useless with no more added value to the business. Therefore, they suggest the use of destruction plans and rules. In case of (Morris, 2018; Möller, 2013), data should be deleted once data that represents the same but is of better quality has become available. All papers that mention the *Destroy* step agree that the destruction of data is an important step to avoid a swamp of outdated or useless data.

3.2 Layers

While iterating over the models found in our SLR, we found four topics, namely *Metadata*, *Data*, *Technology*, and *Enterprise*, that we implemented as layers in DERM. These layers can be seen as verticals, that have an effect on all topics. To illustrate this, we have visualized it using an arrow notation. The following example will explain this effect in more detail.

The European Union passed the General Data Protection Regulation (GDPR) law in 2018. It contains new regulations regarding the digital privacy of European citizens. For instance, the GDPR specifies that for the storage and use of the personal data the consent of the individual must be obtained beforehand. The GDPR topic can be initially placed inside the *Enterprise* layer, as it is a human-made idea about how things should behave within an organization. This leads to a direct impact on the technology level. Tools are now being developed to identify personal data in legacy assets and user interfaces are created in which a user needs to give consent to data processing and storage. This has a direct impact on the data that is gathered and thus on the *Data* layer itself. For example, new data, such as the consent letters, is generated. But also existing data is *transformed*, e.g. to anonymize personal data. Finally, the changes at the *Data* layer have an effect on the *Metadata*. On this layer, it is now possible to attach to each dataset the information whether it contains personal data or not.

Enterprise. The *Enterprise* layer represents the outmost layer in the DERM. It sets general boundaries and conditions for working with data at an organizational level and influences all other layers. The concrete specifications at this layer are often derived from other management efforts within an organization like “IT Management”, “Service Management”, or “Data (Quality) Management” (Hubert Ofner et al., 2013; DAMA, 2017). This layer specifies data governance measures, data standards, specific roles for people working with data (e.g. data owner), and outlines the overall data culture (Hubert Ofner et al., 2013; Elsayed et al., 2008). It also represents aspects that are derived from the general environment an organization operates in, as for example legal or ethical requirements, policies, and administrative concerns (Emam et al., 2019).

Technology. There are several technologies used across the data lifecycle. The *Technology* layer describes these technologies and how they enable the processing of data under consideration of the *Enterprise* guidelines. Most importantly, this includes the specification of data storage technologies like file storage, database systems, data lakes, or archives (Emam et al., 2019; Pääkkönen and Pakkala, 2015). Furthermore, the requirements of underlying technologies, for instance local filesystems, cloud and edge technologies, or P2P networks can be considered in this phase. Additionally, this layer represents the use of other technologies involved in data analytics and management, which can include data quality, data security and privacy, or machine learning tools (Solanas et al., 2017; Cao et al., 2019).

Data. The *Data* layer is the central and most important part of DERM. It is unsurprising that this layer is part of all data lifecycles presented in literature. The differences in literature are in concrete data formats as for example strings, tupel, social networks, graphs, etc. Generally, these data formats can be distinguished between structured, semi-structured, and unstructured data. By iteratively passing through the data lifecycle, a certain data object can take on different formats. For instance, (El Arass et al., 2020) show how data can move from an unstructured raw textfile to a structured knowledge graph by iteratively integrating and updating it with other data objects. Most papers focus on a certain data format for their specific use case, like machine sensor data (Christopherson et al., 2020), biological data (Allanic et al., 2017) or healthcare records (Solanas et al., 2017).

Metadata. The innermost layer of DERM represents the *Metadata* of data objects. The metadata plays an important role in the model as it supports the value and quality of data objects, particularly throughout the *Select/Access* and *Use* phases (Cao et al., 2019). Its specifics are shaped by existing data models and standards in place. Following (W3C, 2014), if a metadate is created, it can support the establishment of data semantics and linked data. The lifecycle of metadata should be closely aligned with the associated data lifecycle to track potential changes at the data or other surrounding layers. To avoid quality issues, metadata should automatically be generated and documented (Maindze et al., 2019). As stated by (Hodge, 2001) “Metadata is key to ensuring that resources will survive and continue to be accessible into the future”, thus making *Metadata* the core of our model.

4 VALIDATION

To answer RQ2 and show that DERM is a valid reference model for the field of data engineering, we tried to position the most common data engineering topics present in literature. To identify those topics, we collected the author keywords of all papers published at the International Conference on Data Engineering (ICDE) from 1997 till 2020. We selected the one hundred most frequent keywords. Keywords were discarded if they were already a part of DERM itself (e.g. *data* or *metadata*) or some kind of duplicate (e.g. *IoT* and *Internet of Things*). We also removed keywords that were out of scope (e.g. *astroparticle physics*) or to fuzzy and broad (e.g. *business* or *computer*).

To place the topics within the model we conducted a focus group discussion with four data engineering professionals (Jarzabkowski, 2008). The positioning is to be understood as a rough direction. Some topics can be added to multiple layers or phases and we have placed them where we believe they play the most significant role. If a topic plays a role in all phases of a layer, we have coded this using *[- -]*. If a topic is significant in all phases and layers, we added it in the middle bar. Our result can be seen in Figure 2.

We were able to categorize all of the topics after a short discussion period. Since each topic could be located as part of the model, we are highly confident that DERM, which is based on data lifecycles found in literature, can be used as a foundation for a reference model in the field of data engineering.

As one can see in Figure 2, there is no equal distribution of topics on the available surface of DERM. Most topics are clustered in the *Use* phase and in its *Data* layer. While the phases *Select/Access* and *Transform* still contain some topics, special topics are underrepresented in the phases of *Plan*, *Create*, and *Destroy*. In general, there are also few topics that address the *Enterprise* layer in the phases.

5 DATA ENGINEERING: A RESEARCH AGENDA

Data engineering is a relatively new discipline in computer science. It is multi-layered and complex, as it encompasses various topics and research directions. (Bosch et al., 2021). To overcome the challenges associated with data engineering, there is a need for further research on the topics. This way the engineering of data-intensive applications can reach the next maturity level and become a more professional discipline.

For answering RQ3 we propose a research agenda that is based on a review of our SLR and the validation of the DERM. Specifically, we reviewed the mapping of the papers in Table 1 and the distribution of topics in Figure 2 to determine what phases and layers have received the least attention so far. For each of these, we see the necessity for further research and formulate concrete research questions.

Using the aforementioned approach, we derived the following distribution of papers and topics to the different phases and layers of our reference model (see Table 2).

Table 2: Distribution of papers and research topics to DERM themes.

	Theme	# papers	# topics
Phases	Plan	14	3
	Create	15	9
	Select/Access	29	18
	Use	31	34
	Transform	24	16
	Destroy	12	1
Layers	Metadata	9	7
	Data	34	51
	Technology	12	20
	Enterprise	11	9

We counted a topic multiple times if it covered more than one phase or layer and was placed crossing the respective boundaries (e.g spatial crowdsourcing). It becomes apparent that the research regarding data lifecycles and in the data engineering community is focused on the same areas. We can conclude that research mostly focuses on certain phases and layers but other subjects are neglected. The same phenomenon applies to real-world projects of engineering data-intensive applications (Kleppmann, 2017; Bosch et al., 2021). Nevertheless, the subjects that have received limited attention so far are important parts within data engineering and should receive additional attention by science and practice.

5.1 Phases

Specifically, from Table 2 and Figure 2, we can derive that the phases *Plan, Create,* and *Destroy* have received significantly less attention as compared to *Access, Use,* and *Transform.* This confirms our own experiences that the engineering of data-intensive applications is focused on the data analysis part and associated steps. For example, questions on legal and ethical guidelines or what happens to data models after they were used are often not considered systematically in the overall data engineering process. We argue that further research is required in these directions and specify the following potential research topics for the DERM phases (see Table 3).

Table 3: Research topics on DERM phases.

Topic	Possible Research Areas
Plan	- Legal and ethical considerations in data engineering - Process and data management aspects for data engineering
Create	- Factors influencing the creation of data - Creating artifical data sets and artificially enlarging data sets - Crowdsourcing as a means of data creation
Destroy	- Handling and decommissioning of data models after their use - Possibilities for re-use in data engineering tasks

5.2 Layers

With regard to the layers that are mentioned in literature, we observe that the research focuses on the *Data* layer, which was mentioned in all SLR papers and has the most associated data engineering topics. This seems logical, since data is the central element of research. However, the layers *Metadata, Technology,* and *Enterprise*, which have an effect on the data layer, are mentioned to a significantly lesser extent. We therefore argue that additional research on the role of these layers in the data engineering process is necessary. Consequently, we propose the following potential research topics for the DERM layers (see Table 4).

6 CONCLUSION

The guiding objective of our study was the development of a reference model for data engineering. The model helps to further professionalize the development of data-intensive applications by offering a common basis for planning and conducting data engineering initiatives. To the best of our knowledge, no model exists that provides a systematic overview of the steps in the data engineering process. Based on our insights, we determined what steps in the engineering process need additional attention and formulated a research agenda for data engineering. We can conclude that we were able to positively answer our proposed research questions and achieved the goal of developing a reference model for data engineering.

Table 4: Research topics on DERM layers.

Topic	Possible Research Areas
Metadata	- Automated generation of metadata from data objects - The mediating role of data catalogs in data collaborations - Automated updates to metadata and versioning of data
Technology	- Technological interaction between different phases in the engineering process - Differences in data engineering on different data technologies (IoT, Social Networks, Blockchain, etc.) - Incorporation of data security and privacy protection aspects - Implementation of data quality guidelines as executable rules
Enterprise	- Roles and Responsibilities in the data engineering process - Data quality management for data engineering - Deriving the provenance of data objects

Our work offers the following **scientific contributions**. Despite the increasing attention data engineering receives from the scientific community, the research seems to concentrate on the usage of data. Other parts within the data engineering lifecycle are often neglected but are important for the success of engineering projects (Bosch et al., 2021). We found that research for the phases *Plan, Create,* and *Destroy* and the layers *Metadata, Technology,* and *Enterprise* is underrepresented and should receive further attention in the future. Therefore, we formulated a set of possible research topics that address unanswered yet important directions for further research.

Building on the aforementioned, we offer **managerial contributions**. There is a need to bridge theory and practice for reaching a higher level of maturity in creating data-intensive applications (Stonebraker and Çetintemel, 2018; Kleppmann, 2017). To achieve this goal, our study offers guidelines for conducting data engineering more profoundly. Such a capability can increase the success of data science initiatives and help to create competitive advantages (Davenport et al., 2006). Specifically, organizations can use DERM to evaluate their internal software and data engineering practices in a systematic way and ensure that all phases and layers are represented. For data science and software engineering teams, DERM can act as a tool that helps to raise the right questions during requirements engineering and the development process and gain a better understanding of the overall requirements.

Despite applying a high level of rigor, our research is subject to several **limitations**. First, our study cannot be free from researcher bias. The paper selection process during the SLR and the validation of our model are subjective and were influenced by the researchers' experiences and backgrounds. Second, the validation of our model is currently based on assigning research topics. It lacks a practical evaluation in the form of an application to a real-world development project.

Based on our findings and limitations, we see promising directions for **future work**. We plan to use our model in different organizational settings to further evaluate its validity. Specifically, we plan to use DERM as part of a requirements engineering workshop in a development project for a machine-learning application. It will hereby act as canvas, where the participants can place the derived requirements and ideas as sticky notes. The feedback from this workshop will help us extend or adapt our model to meet the expectations of software and data engineers. Additionally, we will follow up on some research topics presented in Section 5 to create a deeper knowledge of engineering data-intensive applications within these areas.

ACKNOWLEDGMENTS

This work was funded by the Fraunhofer-Cluster of Excellence »Cognitive Internet Technologies«.

REFERENCES

Alladi, B. S. and Prasad, S. (2018). Big data life cycle: security issues, challenges, threat and security model. *International Journal of Engineering & Technology*, 7(3):100–103.

Allanic, M., Hervé, P.-Y., Pham, C.-C., Lekkal, M., Durupt, A., Brial, T., Grioche, A., Matta, N., Boutinaud, P., Eynard, B., et al. (2017). Biomist: A platform for biomedical data lifecycle management of neuroimaging cohorts. *Frontiers in ICT*, 3:35.

Alshboul, Y., Wang, Y., and Nepali, R. (2015). Big data life cycle: threats and security model. In *Americas conference on information systems*, pages 1–7.

Amadori, A., Altendeitering, M., and Otto, B. (2020). Challenges of data management in industry 4.0: A single case study of the material retrieval process. In *International Conference on Business Information Systems*, pages 379–390. Springer.

Bosch, J., Olsson, H. H., and Crnkovic, I. (2021). Engineering ai systems: a research agenda. In *Artificial Intelligence Paradigms for Smart Cyber-Physical Systems*, pages 1–19. IGI Global.

Bychkov, I., Demichev, A., Dubenskaya, J., Fedorov, O., Haungs, A., Heiss, A., Kang, D., Kazarina, Y., Korosteleva, E., Kostunin, D., Kryukov, A., Mikhailov, A., Nguyen, M.-D., Polyakov, S., Postnikov, E., Shigarov, A., Shipilov, D., Streit, A., Tokareva, V., Wochele, D., Wochele, J., and Zhurov, D. (2018). Russian–german astroparticle data life cycle initiative. *Data*, 3(4):56.

Cao, H., Wachowicz, M., Renso, C., and Carlini, E. (2019). Analytics everywhere: generating insights from the internet of things. *IEEE Access*, 7:71749–71769.

Chen, J., Chen, Y., Du, X., Li, C., Lu, J., Zhao, S., and Zhou, X. (2013). Big data challenge: a data management perspective. *Frontiers of Computer Science*, 7(2):157–164.

Cheng, X., Hu, C., Li, Y., Lin, W., and Zuo, H. (2013). Data evolution analysis of virtual dataspace for managing the big data lifecycle. In *2013 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum*, pages 2054–2063. IEEE.

Christopherson, L., Mandal, A., Scott, E., and Baldin, I. (2020). Toward a data lifecycle model for nsf large facilities. In *Practice and Experience in Advanced Research Computing*, pages 168–175.

DAMA (2017). *DAMA-DMBOK: Data Management Body of Knowledge*. Technics Publications.

Davenport, T. H. et al. (2006). Competing on analytics. *Harvard business review*, 84(1):98.

El Arass, M., Ouazzani-Touhami, K., and Souissi, N. (2020). Data life cycle: Towards a reference architecture. *International Journal of Advanced Trends in Computer Science and Engineering*, 9(4):5645–5653.

El Arass, M., Tikito, I., and Souissi, N. (2017). Data lifecycles analysis: towards intelligent cycle. In *2017 Intelligent Systems and Computer Vision (ISCV)*, pages 1–8.

Elsayed, I., Muslimovic, A., and Brezany, P. (2008). Intelligent dataspaces for e-science. In *WSEAS International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics*, pages 94–100.

Emam, I., Elyasigomari, V., Matthews, A., Pavlidis, S., Rocca-Serra, P., Guitton, F., Verbeeck, D., Grainger, L., Borgogni, E., Del Giudice, G., Saqi, M., Houston, P., and Guo, Y. (2019). Platformtm, a standards-based data custodianship platform for translational medicine research. *Scientific data*, page 149.

Fisher, T. (2009). *The data asset: How smart companies govern their data for business success*, volume 24. John Wiley & Sons.

Grunzke, R., Aguilera, A., Nagel, W. E., Krüger, J., Herres-Pawlis, S., Hoffmann, A., and Gesing, S. (2015). Managing complexity in distributed data life cycles enhancing scientific discovery. In *2015 IEEE 11th International Conference on e-Science*, pages 371–380. IEEE.

Ho, T. and Abramson, D. (2007). Active data: Supporting the grid data life cycle. In *Seventh IEEE International Symposium on Cluster Computing and the Grid (CC-Grid'07)*, pages 39–48. IEEE.

Hodge, G. M. (2001). Metadata made simpler.

Huang, G., Luo, C., Wu, K., Ma, Y., Zhang, Y., and Liu, X. (2019). Software-defined infrastructure for decentralized data lifecycle governance: Principled design and open challenges. In *International Conference on Distributed Computing Systems (ICDCS)*, pages 1674–1683.

Hubert Ofner, M., Straub, K., Otto, B., and Oesterle, H. (2013). Management of the master data lifecycle: a framework for analysis. *Journal of Enterprise Information Management*, 26(4):472–491.

Jarzabkowski, P. (2008). Shaping strategy as a structuration process. *Academy of Management journal*, 51(4):621–650.

Khatri, V. and Brown, C. V. (2010). Designing data governance. *Communications of the ACM*, 53(1):148–152.

Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(2004):1–26.

Kleppmann, M. (2017). *Designing data-intensive applications: The big ideas behind reliable, scalable, and maintainable systems*. " O'Reilly Media, Inc.".

Kuhrmann, M., Fernández, D. M., and Daneva, M. (2017). On the pragmatic design of literature studies in software engineering: an experience-based guideline. *Empirical software engineering*, 22(6):2852–2891.

Levitin, A. V. and Redman, T. C. (1993). A model of the data (life) cycles with application to quality. *Information and Software Technology*, 35(4):217–223.

Levitin, A. V. and Redman, T. C. (1998). Data as a resource: Properties, implications, and prescriptions. *MIT Sloan Management Review*, 40(1):89.

Liu, K., Tan, H. B. K., and Chen, X. (2013). Supporting the adaptation of open-source database applications through extracting data lifecycles. *IET software*, 7(4):213–221.

Maindze, A., Skaf, Z., and Jennions, I. (2019). Towards an enhanced data-and knowledge management capability: A data life cycle model proposition for integrated vehicle health management. *Annual Conference of the PHM Society*, 11.

Miles, M. B. and Huberman, A. M. (1994). *Qualitative data analysis: An expanded sourcebook*. sage.

Moiso, C., Antonelli, F., and Vescovi, M. (2012). How do i manage my personal data? - a telco perspective. In *DATA*, pages 123–128.

Möller, K. (2013). Lifecycle models of data-centric systems and domains. *Semantic Web*, 4(1):67–88.

Morris, C. (2018). The life cycle of structural biology data. *Data Science Journal*, 17:26.

Otto, B. (2015). Quality and value of the data resource in large enterprises. *Information Systems Management*, 32(3):234–251.

Pääkkönen, P. and Pakkala, D. (2015). Reference architecture and classification of technologies, products and services for big data systems. *Big Data Research*, 2(4):166–186.

Polyzotis, N., Roy, S., Whang, S. E., and Zinkevich, M. (2018). Data lifecycle challenges in production machine learning: a survey. *ACM SIGMOD Record*, 47(2):17–28.

Rahul, K. and Banyal, R. K. (2020). Data life cycle management in big data analytics. *Procedia Computer Science*, 173:364–371.

Simonet, A., Fedak, G., Ripeanu, M., and Al-Kiswany, S. (2013). Active data: a data-centric approach to data life-cycle management. In *Proceedings of the 8th Parallel Data Storage Workshop*, pages 39–44.

Sinaeepourfard, A., Garcia, J., Masip-Bruin, X., and Marín-Tordera, E. (2016). A comprehensive scenario agnostic data lifecycle model for an efficient data complexity management. In *International Conference on e-Science (e-Science)*, pages 276–281. IEEE.

Solanas, A., Casino, F., Batista, E., and Rallo, R. (2017). Trends and challenges in smart healthcare research: A journey from data to wisdom. In *2017 IEEE 3rd International Forum on Research and Technologies for Society and Industry (RTSI)*, pages 1–6. IEEE.

Stonebraker, M. and Çetintemel, U. (2018). " one size fits all" an idea whose time has come and gone. In *Making Databases Work: the Pragmatic Wisdom of Michael Stonebraker*, pages 441–462.

Strauss, A. and Corbin, J. M. (1997). *Grounded theory in practice*. Sage.

Tallon, P. P., Ramirez, R. V., and Short, J. E. (2013). The information artifact in it governance: toward a theory of information governance. *Journal of Management Information Systems*, 30(3):141–178.

Tripathi, D. and Pandy, S. R. (2018). Developing a conceptual framework of research data management for higher educational institutions. In *2018 5th International Symposium on Emerging Trends and Technologies in Libraries and Information Services (ETTLIS)*, pages 105–110. IEEE.

W3C (2014). Best practices for publishing linked data.

Webster, J. and Watson, R. T. (2002). Analyzing the past to prepare for the future: Writing a literature review. *MIS quarterly*, pages 13–23.

Wing, J. M. (2019). The data life cycle. *Harvard Data Science Review*, 1(1).

Xianglan, L. I. (2017). Digital construction of coal mine big data for different platforms based on life cycle. In *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)(*, pages 456–459. IEEE.

Yazdi, M. A. (2019). Enabling operational support in the research data life cycle. In *Proceedings of the First International Conference on Process Mining*, pages 1–10.

Yu, X. and Wen, Q. (2010). A view about cloud data security from data life cycle. In *2010 international conference on computational intelligence and software engineering*, pages 1–4. IEEE.