A Design Theory for Data Catalogs

Dissertation

zur Erlangung des Grades eines

D o k t o r s d e r I n g e n i e u r w i s s e n s c h a f t e n

der Technischen Universität Dortmund
an der Fakultät für Informatik

von

Daniel Tebernum

Dortmund

2024

Tag der mündlichen Prüfung: 09.10.2024
Dekan: Prof. Dr. Jens Teubner
Gutachter: Prof. Dr. Falk Howar & Prof. Dr.-Ing. Frederik Möller

Abstract

In today's data-driven world, individuals, companies, and government agencies are generating and collecting enormous amounts of data at an increasing rate. This vast amount of data offers immense potential for valuable insights, informed decision-making, and value creation. The correct data can help optimize processes, make predictions, or establish new business models. To exploit this potential, professional and modern data management that supports data discovery, data governance, and data democratization is essential. For this purpose, data catalogs are a valuable tool. They act as a centralized repository within an organization or institution, allowing users to discover, understand, and access data quickly.

Data catalogs are gaining popularity in many fields, but holistic, practice-based, and design-oriented knowledge is still lacking. Thus, the goal of this thesis is to provide a design theory that will aid scholars and professionals in the process of designing data catalogs. As a basis for developing the design theory, we utilized our data catalog, DIVA, which we developed over several iterations and years in close exchange with practice. We have done this to create a design theory grounded in practice that is relevant to both researchers and practitioners. Prescriptive design knowledge was extracted from DIVA in the form of design principles. Concrete recommendations for action in the form of design features were also developed based on DIVA. In a qualitative study, people from the target group of our design theory evaluated the results.

We present design knowledge for data catalogs of different maturity levels. On the one hand, implicit design knowledge is given as software artifacts. Further, design knowledge has been published in the form of models, methods, and architectures in peer-reviewed publications, which are part of this thesis. Mainly, this work deals with the development of design principles and design features. In sum, the contributions compose a design theory for data catalogs.

This thesis contributes to the body of design-oriented knowledge concerning data catalogs and thus also data management in general. The design theory is intended to support researchers and practitioners in designing or developing successful data catalogs. It aids them by providing prescriptive design knowledge and concretizing examples from practice and literature.

Keywords: data catalogs, design knowledge, design principles, design features, design theory

Contents

1 Introduction **1**
- 1.1 Motivation and Problem Statement . . . 1
- 1.2 Research Questions . . . 4
- 1.3 Structure of the Thesis . . . 5

2 Background **7**
- 2.1 Metadata . . . 7
  - 2.1.1 Metadata Types . . . 8
  - 2.1.2 Metadata Models . . . 8
  - 2.1.3 Metadata Challenges . . . 9
- 2.2 Data Catalogs . . . 9
  - 2.2.1 Historical Context of Data Catalogs . . . 10
  - 2.2.2 Definition of Data Catalogs . . . 10
  - 2.2.3 Data Catalog Types . . . 12
  - 2.2.4 Data Catalog Solutions Relevant to the Thesis . . . 15
  - 2.2.5 Additional Terminology Related to Data Catalogs . . . 16

3 Related Work **19**
- 3.1 Generating the Data Catalog Corpus . . . 19
- 3.2 Discussion of the Data Catalog Corpus . . . 20

4 Research Design **23**
- 4.1 Overview of the DSR Project . . . 23
- 4.2 Development of the Design Theory . . . 25

5 DIVA Development **29**
- 5.1 Iteration 1 . . . 29
- 5.2 Iteration 2 . . . 31
- 5.3 Iteration 3 . . . 33
- 5.4 Iteration 4 . . . 35
- 5.5 Iteration 5 . . . 37
- 5.6 Iteration 6 . . . 39
- 5.7 Iteration 7 . . . 41
- 5.8 Iteration 8 . . . 44
- 5.9 Summary . . . 46

6 Designing Data Catalogs 49
6.1 DP1: Principle of Automation . . . 49
6.1.1 DF1.1: Automated Inventory . . . 50
6.1.2 DF1.2: Automated Metadata Gathering . . . 51
6.2 DP2: Principle of Flexibility . . . 54
6.2.1 DF2.1: Extensible Metadata Model . . . 55
6.2.2 DF2.2: Creation and Customization of Metrics . . . 56
6.2.3 DF2.3: Customizable Policy System . . . 58
6.2.4 DF2.4: Configurable Workflow Engine . . . 59
6.3 DP3: Principle of Interoperability . . . 60
6.3.1 DF3.1: Standardized Metadata Models . . . 62
6.3.2 DF3.2: Standardized API Documentations . . . 63
6.3.3 DF3.3: Open Source . . . 64
6.4 DP4: Principle of Context . . . 65
6.4.1 DF4.1: Metadata With Contextual Information . . . 67
6.4.2 DF4.2: Data Networks . . . 68
6.5 DP5: Principle of Data Life Cycle Management . . . 69
6.5.1 DF5.1: Data Usage Policies . . . 70
6.5.2 DF5.2: End-of-Life Data Management . . . 71
6.6 DP6: Principle of Visualization . . . 73
6.6.1 DF6.1: Metadata Visualization . . . 74
6.6.2 DF6.2: Data Network Visualization . . . 75
6.7 DP7: Principle of Data Assessment . . . 76
6.7.1 DF7.1: Quality Metrics . . . 78
6.7.2 DF7.2: Review System . . . 79
6.7.3 DF7.3: Risk Management Capabilities . . . 79
6.8 A Design Theory for Data Catalogs . . . 80

7 Evaluation 85
7.1 Evaluation Design . . . 85
7.2 Individual Evaluation of the Design Principles . . . 89
7.2.1 Principle of Automation . . . 89
7.2.2 Principle of Flexibility . . . 90
7.2.3 Principle of Interoperability . . . 91
7.2.4 Principle of Context . . . 92
7.2.5 Principle of Life Cycle Management . . . 93
7.2.6 Principle of Visualization . . . 94
7.2.7 Principle of Data Assessment . . . 95
7.3 Overall Evaluation of the Design Principles . . . 96
7.3.1 Evaluation of the Questionnaire . . . 96
7.3.2 Comparison of the Questionnaire and the Expert Discussions . . . 99
7.3.3 Discussion of Possible Deficiencies . . . 101
7.3.4 Overarching Expert Comments . . . 102
7.4 Discussion . . . 103

8 Conclusion **105**
8.1 Answers to the Research Questions . . . 105
8.2 Limitations . . . 108
8.3 Future Work . . . 111

Bibliography **113**

Data Catalog Corpus **135**

Appendix **147**

A DIVA Screenshots **147**

B Evaluation Infomaterial **165**

C Evaluation Questionnaire **169**

D Expert Discussion Notes **171**
D.1 Expert E1 . . . 171
D.2 Expert E2 . . . 174
D.3 Expert E3 . . . 176
D.4 Expert E4 . . . 179
D.5 Expert E5 . . . 181
D.6 Expert E6 . . . 183
D.7 Expert E7 . . . 185
D.8 Expert E8 . . . 188

E Own Publications **193**

Acknowledgments **197**

List of Figures

2.1 Evolution of data documentation [Kor19; Sen04b] 11

4.1 Overview of our DSR project 24
4.2 Classification of the design knowledge for data catalogs using the positioning framework of Wache et al. [Wac22] 27

5.1 Summary of the DIVA development iteration 1 29
5.2 Summary of the DIVA development iteration 2 32
5.3 Summary of the DIVA development iteration 3 34
5.4 Summary of the DIVA development iteration 4 36
5.5 Summary of the DIVA development iteration 5 37
5.6 Summary of the DIVA development iteration 6 39
5.7 Summary of the DIVA development iteration 7 41
5.8 Summary of the DIVA development iteration 8 44

6.1 Positioning of the DIVA workflow engine implementations in the classification dimensions according to Stojnić [Sto15] 52
6.2 Levels of conceptual interoperability by Wang et al. [Wan09] 61

7.1 Assessment criteria of the DPs expressiveness [Jan20] 87
7.2 DP1 expert discussion evaluation results 89
7.3 DP2 expert discussion evaluation results 90
7.4 DP3 expert discussion evaluation results 91
7.5 DP4 expert discussion evaluation results 92
7.6 DP5 expert discussion evaluation results 93
7.7 DP6 expert discussion evaluation results 94
7.8 DP7 expert discussion evaluation results 95
7.9 Results of the questionnaire 97
7.10 Aggregated questionnaire results 100
7.11 Aggregated expert discussion results 100

8.1 Overview of the mapping of DPs, DFs, and peer-reviewed artifacts 108

A.1 DIVA 1.0: Search functionality of the data catalog 147
A.2 DIVA 1.0: Details view 147
A.3 DIVA 1.0: Data inventory form 148
A.4 DIVA 1.0: Data inventory business form 148
A.5 DIVA 1.0: Data inventory price form 149

A.6 DIVA 1.0: Data evaluation questionnaire . . . 149
A.7 DIVA 1.0: Management dashboard . . . 150
A.8 DIVA 1.1: Custom metrics . . . 151
A.9 DIVA 1.1: Data source evaluation results . . . 152
A.10 Data Asset Crawler screenshots (1) . . . 153
A.11 Data Asset Crawler screenshots (2) . . . 154
A.12 DIVA 2.1: Metadata representation of tabular data . . . 155
A.13 DIVA 3.0: Metadata representation of tabular data . . . 155
A.14 DIVA 3.0: Visualization of the data network . . . 156
A.15 DIVA 3.0: Data Space Connector usage policy configuration . . . 156
A.16 DIVA 4.0: Details view . . . 157
A.17 DIVA 4.0: A form for creating new metadata fields . . . 158
A.18 DIVA 4.0: Data network visualization . . . 159
A.19 DIVA 4.0: Metadata representation of tabular data . . . 160
A.20 DIVA 4.0: Management dashboard . . . 161
A.21 DIVA 4.0: Review system . . . 161
A.22 DIVA 4.1: Destroy Claim overview . . . 162
A.23 DIVA 4.1: Destroy Claim details part 1 . . . 163
A.24 DIVA 4.1: Destroy Claim details part 2 . . . 163

List of Tables

2.1 Typology of data catalogs [Jah23] 13

4.1 Components of the DP schema based on Gregor et al. [Gre20] and adapted to own needs 28

6.1 DP1: Principle of Automation 49
6.2 DP2: Principle of Flexibility 54
6.3 DP3: Principle of Interoperability 60
6.4 DP4: Principle of Context 66
6.5 DP5: Principle of Data Life Cycle Management 69
6.6 DP6: Principle of Visualization 73
6.7 DP7: Principle of Data Assessment 77

7.1 Characterization of the participants 87

D.1 Discussion notes E1 171
D.2 Discussion notes E2 174
D.3 Discussion notes E3 176
D.4 Discussion notes E4 179
D.5 Discussion notes E5 181
D.6 Discussion notes E6 183
D.7 Discussion notes E7 185
D.8 Discussion notes E8 188

Acronyms

AI	Artificial Intelligence
API	Application Programming Interface
AWS	Amazon Web Services
CKAN	Comprehensive Knowledge Archive Network
CMS	Content Management System
DAC	Data Asset Crawler
DAI	DIVA Artifact Iteration
DCA	Destroy Claim Agent
DCAT	Data Catalog Vocabulary
DCAT-AP	DCAT Application Profile
DCE	Dublin Core Metadata Element Set
DCTERMS	Dublin Core Metadata Initiative Metadata Terms
DF	Design Feature
DIVA	Data Inventory and Valuation Approach
DP	Design Principle
DSC	Dataspace Connector
DSR	Design Science Research
EDC	Eclipse Dataspace Components
EU	European Union
FaaS	Functions as a Service
FAIR	Findable, Accessible, Interoperable, and Re-usable
FOSS	Free and Open-Source Software
Fraunhofer ISST	Fraunhofer Institute for Software and Systems Engineering ISST
GDPR	General Data Protection Regulation
HIPAA	Health Insurance Portability and Accountability Act
HTTP	Hypertext Transfer Protocol
IDS	Industrial Data Space
IoT	Internet of Things
IPFS	InterPlanetary File System
ISO	International Organization for Standardization

JACPoL	JSON-Based Access Control Policy Language
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LSH	Locality-Sensitive Hashing
M4DG	Metadata Model for Data Goods
MIME	Multipurpose Internet Mail Extensions
MRQ	Main Research Question
NASA	National Aeronautics and Space Administration
NISO	National Information Standards Organization
ODP	Open Data Portal
ODRL	Open Digital Rights Language
OGD	Open Government Data
PII	Personally Identifiable Information
RAKE	Rapid Automatic Keyword Extraction
RDF	Resource Description Framework
REST	Representational State Transfer
RoI	Return on Investment
RQ	Research Question
SLR	Systematic Literature Review
SMD	Science Mission Directorate
TP	Testable Proposition
WWW	World Wide Web

CHAPTER 1

Introduction

> "Imagine managing books without title information, author data, cover images, royalties, or number of chapters. That's what it's like managing data without a catalog. [...]"
> — Rupal Sumaria, Head of Data Governance at Penguin Random House UK [Sum24]

Data catalogs have become integral to modern data management [Lab20b; Sha16c]. They enable data discovery [Ole23], support the implementation of the Findable, Accessible, Interoperable, and Re-usable (FAIR) principles [Bor22; Lab20b; Qui20], promote data democratization [Eic22a; Lef21] and contribute to the implementation of data governance [Che22; Rya22; Sha16c]. Despite their significant importance, implementing data catalogs remains a challenge in practice [Lab20b]. Numerous research projects aim to improve the knowledge about data catalogs, be it in the context of open data portals [Ade17; Kir19c; Kře19; Urb22], enterprise data catalogs [Bug22; Ehr21a; Jah23; Lab20a; Sha16b], or data marketplaces [Azc22; Eic22a; Eic22b]. However, a lack of knowledge about the characteristics of data catalogs makes it challenging to implement them successfully in practice [Jah23]. In this chapter, we justify our research by emphasizing the relevance of data catalogs in modern data management and the current lack of practice-based, design-oriented knowledge. We then delve into our research questions, which are the cornerstone of our work and will guide our discussion.

1.1 Motivation and Problem Statement

Digital transformation is an important driver to stay competitive in today's business world [Cal24]. Especially data, seen as an asset, can give organizations a competitive advantage [Leg17]. Data-driven decisions are a core feature to meet the market's growing and dynamic demands and guarantee customer satisfaction [Szu23]. So, data is seen as more than just pure information. For companies, data is not just a resource but a strategic asset that must be effectively managed for success [Ott15; Ott07; Tal14]. So, it is no surprise that statements like "data is the commodity of the 21st century" [Mer16] can be heard more often when discussing digitization. As the use of data continues to grow, there is a pressing need for robust data management solutions that adhere to the FAIR principles, data governance, and data democratization [Sam22]. All three concepts are interconnected and aim to generate value from data.

Enhancing the findability, accessibility, interoperability, and reusability of data is the goal of the **FAIR principles** [Jac20]. These guidelines were initially promoted for better

data exploitation in the research community [Wil16]. The intention is to make data more machine- and human-readable and shareable, boosting scientific research and innovation productivity [Axt16]. The ideas of FAIR data are being embraced by more and more areas to encourage transparency and openness in research. For example, enterprises have also recognized the potential of FAIR to make more use of their data [Lab20b]. The FAIR principles can be seen as a first step when it comes to making use of data. However, "[...] even though the principles create a powerful platform for furthering data sharing and improving data stewardship, they do not address the normative issues and challenges associated with data sharing." [Boe18, p. 931].

Due to this, many companies are also concerned with the topic of **data governance**. Data governance deals with the overall management of organizational data and questions of control over the data and what responsibilities there are [Int17]. "Data Governance represents the leading function of data management as it specifies which decisions need to be made in data management and who makes these decisions. Data management ensures these decisions are made and appropriate action takes place." [Ott11]. Data governance is seen as a critical factor in increasing data value and minimizing risks [Abr19].

Alongside the FAIR principles and data governance, companies are also trying to establish a data-driven culture by implementing **data democratization**. Data democratization is the "[...] act of opening organizational data to as many employees as possible, given reasonable limitations on legal confidentiality and security" [Awa20, p. 1]. This is done, because not the data itself or specific individuals create value, but the fact that many people can access and use the data [Zen18]. The focus is on improving the collaboration between different organizational units, roles, and people. Employees should be able to find suitable data for their tasks autonomously. "Further, [data democratization] represents the organizational understanding of FAIR" [Sam22, p. 1]. The same or at least similar challenges can also be found in government agencies that want or even have to share data with their citizens.[1] In that case, data democratization is called open data [Awa20].

Efforts are being made to implement these concepts to increase value creation using data, but three major obstacles are hampering progress. First, **exponential data growth**. More and more data is being generated faster in many domains like finance, health, government, and social media [Ryd18]. The trend is fueled by many technologies, such as the Internet of Things (IoT), cloud computing, and the general availability of devices like smartphones. Managing and working with data effectively and extracting relevant information for decision-making is becoming increasingly difficult [Rod16].

Second, the strong **heterogeneity** of the data landscape is also a challenge [Kum21]. Due to the many different persistence technologies, transmission protocols, and data formats, data cannot simply be searched across the board. There is an almost infinite amount of different types of data. To find suitable data, one would need to possess a deep understanding and be able to handle an almost endless amount of technical and content-related constellations.

Third, another severe problem involves **data silos**. Data silos are isolated data-holding

1 `https://digital-strategy.ec.europa.eu/en/policies/strategy-data` [Accessed: May 23, 2023]

repositories that few employees in the enterprise know about [Car22]. Opening them up could provide valuable insights to other employees. The larger the company, the more data there is, the more likely these data silos will form. For example, departments collect and store data only for their tasks and interests on internal departmental data platforms. Therefore, there is no chance to find and retrieve the data from outside. Companies are aware of the problem and are trying to bridge these silos in a costly manner through integration strategies [Pat19].

In sum, these three obstacles lead to data scientists and knowledge workers spending a significant amount of time searching for data. They are assumed to spend 25% [Ros10b] to 98% [Den17] of their working time searching for appropriate data. Implementing the FAIR principles, data governance, data democratization, and overcoming these three challenges calls for advanced technological solutions that significantly improve data management.

Data catalogs are a recent development that can help support the concepts and address the challenges discussed above [Zai17]. Making data findable, known as data discovery, is their main purpose [Ole23]. They mainly do this by maintaining metadata about data and making it searchable for users. This requires the data catalog to provide adequate metadata that can be used to answer the user's query. In this way, they can also support the FAIR principles [Lab20b; Wil16], data governance [Che22; Rya22; Sha16c], and data democratization [Eic22a; Lef21]. In all three concepts, it is a question of the appropriate metadata that allows users to, e.g., find, use, regulate, or share data. By storing metadata about data, data catalogs are also not directly affected by exponential data growth, data heterogeneity, and data silos. Metadata is a separate layer that provides a uniform depiction of data and enables data management without physically interacting with the data. However, collecting metadata for these purposes is not trivial and requires professional planning and implementation. Collecting metadata can be costly [Jef20], a security risk [Gou22], or result in *big metadata* that is difficult to search [Bra17].

It should come as no surprise that there are still a lot of difficulties and unresolved questions in practice and research regarding data catalogs [Ehr21b; Nik21; Por20; Won23]. Issues that can hinder the usage of data catalogs include poor metadata quality, a lack of multilingualism, documentation, visualization, statistics, user ratings, or the absence of an interconnectivity standard. There are also reports on failed implementations of data catalogs for various reasons. For example, a poorly functioning search function or missing features can render the data catalog useless [Pru21]. Also, problems with unregulated access, unintuitive use, granular changes to the business that cannot be mapped in the data catalog, or needing to assess the consequences of data use are cited as reasons for failed data catalog projects [Str24]. Overall, there appears to be a need for more understanding of the fundamental building blocks of data catalogs [Eic21].

Therefore, this thesis provides holistic, practice-based, and design-oriented knowledge. Through this, "[. . .] important lessons learned both in research and practice can be captured and shared widely, with consequent valuable benefits to industry and society." [Gre13b, p. 2]. When we look at existing literature, especially when it comes to general, verified knowledge about data catalogs, the body of publications is getting thin [Ehr21b; Jah23; Lab20b; Sha16c]. Numerous papers discuss different aspects of data catalogs in detail but do not offer a holistic, practice-based, and design-oriented perspective on how data

catalogs should be constructed. To the best of our knowledge, there is no compiled and structured form of this kind of knowledge for data catalogs.

1.2 Research Questions

As previously discussed, data catalogs represent a significant technical solution for supporting FAIR principles, data governance, data democratization, and data discovery. Despite their potential, only a limited amount of research focuses on data catalogs in a holistic manner [Ehr21b; Jah23; Lab20b; Sha16c]. In particular, the essential question of how data catalogs should be designed to be successful in practice still needs to be answered. To the best of our knowledge, no existing works compile and refine accumulated practice-based, design-oriented knowledge concerning the holistic design of data catalogs. Considering this, our research is guided by the following Main Research Question (MRQ):

MRQ: *How to design data catalogs?*

We need to collect appropriate design knowledge to answer this question. Design knowledge is essential for implementing a particular type of system or solution [Jon07]. It contains knowledge about how artifacts are composed and created [Gre13a]. The research stream concerned with generating design knowledge is called design science [Ake04]. Design knowledge can exist in many different forms. Examples include models, methods, architectures, software instantiations, solution instances, design principles, or design theories [Vai04].

To make design knowledge reusable, it should follow a structured form [Cha15b; Gre20; Jon07; Run20]. Software instantiations alone are unsuitable for this purpose, for example, because they are implemented very situationally and often contain design knowledge only implicitly [Gre13a]. On the other side of the spectrum, design theories represent the most mature and structured form of design knowledge [Gre13a; Jon07]. Therefore, we want to answer the MRQ with a design theory. To generate a design theory for data catalogs, we must also answer the two following Research Questions (RQs).

RQ1: *What are design principles for data catalogs?*

Design Principles (DPs) are the main component of a design theory [Hei14; Jon07]. DPs are a way to structure knowledge and thus make it reusable for others [Kru16]. They "[...] are design decisions and design knowledge that are intended to be manifested or encapsulated in an artifact, method, process or system." [Gre02, p. 17]. DPs "[...] capture the knowledge gained about the process of building solutions for a given domain, and encompass knowledge about creating other instances that belong to this class [...]" [Sei11, p. 45]. In this case, the term "others" refers to the target group of DPs. The target group includes "those who implement systems and those who enact them" [Gre20, p. 1636]. DPs are artifacts that may emerge as a result of a Design Science Research (DSR) project. However, since they act as blueprints to create other artifacts, they are also referred to as meta-artifact [Iiv03].

An essential aspect of creating DPs is knowledge abstraction. "[A]bstraction refers to the process of deriving abstract concepts (e.g., generic features) from observed instances

[...]" [Gre13b, p. 2]. To "[...] optimize rigor, novelty, and relevance of reported research, the researcher should strive to express the technological rule [(design principles)] at the highest useful abstraction level [...]" [Eng20, p. 2635]. Therefore, a DP should not only apply in specific situations but "must be applicable to a class of problems." [Kru16, p.39]. This high degree of abstractness also accounts for their proximity to design theories. Because a theory "[...] is said to arise from identifying the key links between data and prescriptions (or propositions) by discarding detailed information, and the abstraction is a process of deriving key concepts observed in a specific instance [...]" [Lee11, p. 7].

For this thesis, this means that not only specific data catalogs in a fixed context can benefit from the design knowledge, but the whole class of systems of data catalogs in general. We extracted DPs from an existing data catalog software that we developed over more than six years of industry and research initiatives to achieve practice grounding.

> **RQ2:** *What are design features that support the concrete implementation of our design principles?*

The abstractness of the DPs limits their direct application and transfer into a concrete solution instance. This can be problematic if the fundamental reasoning behind a DP can be understood, but how to implement it is unclear [Kru16]. In contrast, Design Features (DFs) are not formulated as abstractly as DPs and can thus be more easily translated into concrete implementations in practice. They "are specific artifact capabilities to satisfy DPs" [Met15, p. 814] and "[...] represent specific ways to implement design principles in an actual artifact [...]" [Rhy17]. Like DPs, they can be represented in a structured and prescriptive form [Fu15]. However, they are formulated much more concretely than DPs, facilitating design knowledge application. DFs can complement DPs while also improving an overall design theory [Jon07]. We particularly want to address practical relevancy and grounding, so we consider the provision of DFs elementary.

DFs are generated in this work by concretizing the DPs. Here, concretizing means determining which function in a data catalog directly supports the implementation of the DP. In this process, the DFs are constrained by the data catalog software we used to extract the DPs. This ensures that only practice-relevant DFs are proposed.

1.3 Structure of the Thesis

Chapter 2 outlines the necessary knowledge to comprehend this thesis fully. Essential concepts such as metadata and data catalogs are presented and discussed here. Chapter 3 presents the state of the art in data catalogs. Here, we look at existing work that provides design knowledge for data catalogs. In particular, existing knowledge in the form of DPs is discussed. In the following Chapter 4, we discuss our research methodology. We discuss how we structured our overall DSR process and how we developed our design knowledge in the form of DPs, DFs, and the design theory for data catalogs.

In Chapter 5, we show how our data catalog DIVA, used to extract practice-based design knowledge for our design theory, was developed. We also collect all essential results obtained during the development of DIVA, which will be the basis for abstracting DPs. In Chapter 6, we present our results in the form of DPs, DFs, and a design theory. Our results

from Chapter 6 are assessed in Chapter 7. Here, we describe how we evaluated our design knowledge in collaboration with practitioners and members of this thesis's target group. We also describe the results in detail and discuss them. Finally, in Chapter 8, we conclude by answering the research questions, discussing limitations, and presenting future work.

CHAPTER 2

Background

This chapter aims to provide fundamental knowledge necessary for comprehending the subsequent work. We start with an introduction to metadata, followed by an in-depth examination of data catalogs, a specialized technical solution from the field of data management, and the main topic of this thesis.

2.1 Metadata

"Metadata [...] describes a discipline that fosters the study of data about data." [Sen04b, p. 1]. Metadata, especially concerning data catalogs, is a crucial primary topic whose knowledge is vital for understanding this thesis. Therefore, in the following, we will discuss metadata, its use, how it can be categorized, and the challenges involved.

Metadata serves as a powerful tool, enabling us to gain deeper insights and a more comprehensive understanding of data, all without the need for direct interaction with the data [Sab22]. Whenever data is generated, whether it's measurements, audio, movies, or any other form, metadata is always there, either implicitly or explicitly. Metadata plays a crucial role in providing context to the data. It's what tells us the unit of the values in measurements, the artist of an audio piece, or the camera used to record a film. Also, information about the size, format, license, or data scheme is conceivable. Often, metadata is also referred to as "data about data" in a simplified way. A more comprehensive definition of how we understand metadata in this thesis comes from the National Information Standards Organization (NISO).

> "Metadata is structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use, or manage an information resource."
> — NISO [Org04, p. 1]

Whether metadata is seen as metadata or data depends very much on the context of use. If the metadata is the actual payload on which analyses are performed, for example, then metadata can be regarded as data at that moment. "The only difference between metadata and data is a mode of use" [Jef20, p. 126].

Having, maintaining, and managing sufficiently good metadata means that companies can improve their **productivity**, **compliance**, and **scalability** [Ros10b]. An increase in **productivity** can be expected, for example, by the fact that data for one's project can be found more quickly. It should be noted that metadata can be used for more than just finding data. Productivity can also be increased as metadata "[...] is also for contextualisation (e.g. relevance, quality, restrictions (rights and costs)) and for coupling users, software and computing resources to data [...]" [Jef20, p. 126]. The improvement of **compliance** can be

expected because good metadata can be used to define the operational requirements clearly. Data can be subject to legal regulations that must be complied with. For example, the General Data Protection Regulation (GDPR) [Eur16] or the Health Insurance Portability and Accountability Act (HIPAA) [Act96], which impose precise requirements on the use of data. If metadata reflects this, appropriate mechanisms can be implemented to prevent, for example, the use of data outside its intended purpose. The third point, **scalability**, is related in the context of metadata, among other things, to the fact that much more data can be processed since it no longer has to be accessed and viewed directly to gain insight into the content.

Finally, metadata can be implicit or explicit. As an example, this can be demonstrated by looking at the dimensions of a digital image. This information can be given explicitly in the file as the number of pixels in height and width. However, it is also implicitly present in the image and can be actively calculated by counting pixels.

2.1.1 Metadata Types

It is necessary to briefly discuss the existence of different types of metadata to further understand the thesis. One can classify metadata into different categories depending on the content statement. Several works have dealt with this categorization. For example, in the field of data lake research, Diamantini et al. have categorized data into *business metadata*, *technical metadata*, and *operational metadata* [Dia18]. In describing Linked Data Sets, metadata was divided into *general metadata*, *access metadata*, and *structural metadata* by Alexander et al. [Ale11]. For this thesis, we will stick to the work of Riley [Ril17]. She also divides metadata into three categories, which will be briefly explained below.

Descriptive metadata describes all forms of metadata that describe the data and thus either make it easier to find or increase understanding about the data. This description includes titles, descriptions, subject fields, provenance, keywords, and the like.

Administrative metadata includes everything needed to manage and use the data. Riley further divides this item into subcategories. It is sufficient to know that this includes information like creation date, checksum, copyright, responsibilities, or location.

Structural metadata describes the form of the data. For text, this could include a table of contents. For table data, the schema of the table. In general, structural metadata is information that helps to navigate the data.

2.1.2 Metadata Models

An important goal when working with metadata is to achieve a common understanding of its structure and meaning. “Metadata must be machine-understandable as well as human-understandable [...]” [Jef20, p. 127]. Only then is it possible to properly understand metadata and ensure interoperability. For this reason, many metadata standards have been formed over time in different domains [Bro03; Eic20; Kim99; Rah12]. These models implement both aspects: standardizing the structure and creating a uniform understanding of the metadata. The following will discuss the most important models used in data catalogs.

One of the best-known standardized metadata models is the Dublin Core Metadata Element Set (DCE) [Wei98], which was published in February 2009 by the Dublin Core

Metadata Initiative as ISO Standard 15836[1]. It standardizes basic properties such as *creator*, *format*, *language*, or *title*.[2] The vocabulary includes 15 properties and is extended by the Dublin Core Metadata Initiative Metadata Terms (DCTERMS)[3] with additional properties. Due to its simplicity, comprehensibility, flexibility, and interoperability, DCE serves as a basis for other metadata models [Ala09; Ale09; Maa10b; Tam07]. Based on DCTERMS, Maali et al. developed the Data Catalog Vocabulary (DCAT) [Maa10b]. They compared several existing metadata models from the field of data catalogs and developed a vocabulary for recurring properties. It no longer focuses on a general description of resources but explicitly deals with the standardized description of data sets. The goal of DCAT is to improve the interoperability of different data catalogs and to simplify the search for data sets. Over time, DCAT has been further improved based on real-world experience [Alb23] and is currently available in Version 3[4]. DCAT is the basis for many other metadata models for data catalogs. It regularly serves as the basis for specializations, for example DCAT Application Profile (DCAT-AP)[5], a widely used application profile for Open Data Portals (ODPs) in the European Union (EU).

2.1.3 Metadata Challenges

Metadata also introduces challenges that need to be addressed. As the amount of data grows, so does the amount of metadata. However, "[m]etadata collection is expensive so incremental collection along the workflow is required" [Jef20, p. 128]. Collection requires mature solutions for comprehensive, efficient, and scalable metadata management and automated extraction and generation of metadata. Aspects such as privacy must also be addressed. Metadata can contain sensitive information, like user data, that must be removed or protected. For example, pictures taken with a smartphone often contain location information that can be misused by third parties [Gou22]. Finally, there is the issue of metadata quality. Just like data, metadata can be of poor or sound quality. Poor quality metadata can be a big problem since, as a consequence, its functions are impaired. The actual data becomes difficult or impossible to find or interpret. Generally speaking, good metadata is needed, and "[. . .] it is often accepted that good metadata is metadata that can answer the 5 W's: Who, What, Where, When and Why." [Qui20, p. 141]. Due to the ever-increasing mass of metadata, this is an issue where automated quality checking is crucial [Och09].

2.2 Data Catalogs

Data catalogs "collect, create and maintain metadata" [Qui20, p. 141]. The collection of metadata in a data catalog is also referred to as building an inventory of data [Jah23; Zai17]. Users can then use the data catalog's search interface to search and find data. "Searching and actually finding data is called data discovery, and that's what a data catalog

1 `https://www.iso.org/standard/52142.html` [Accessed: July 14, 2023]
2 `https://www.dublincore.org/specifications/dublin-core/dces/` [Accessed: July 14, 2023]
3 `https://www.dublincore.org/specifications/dublin-core/dcmi-terms/` [Accessed: July 14, 2023]
4 `https://www.w3.org/TR/vocab-dcat-3/` [Accessed: July 14, 2023]
5 `https://joinup.ec.europa.eu/collection/semantic-interoperability-community-semic/solution/dcat-application-profile-data-portals-europe` [Accessed: July 14, 2023]

is all about." [Ole23, p. 13]. Data catalogs can be seen as a link between data supply and demand [Jah23].

Data discovery is needed to find the best data for one's projects [Ole23]. Without a way to find existing data, many data-driven projects would only rely on the data already known to the project's leaders. There is a risk that important data relevant to the project will be overlooked. Furthermore, it is essential to note that data discovery means searching *for* data and not searching *in* data [Ole23]. Searching in data means, that we already have data and try to answer questions with it. Searching for data means trying to find a source or location where data is stored. Data catalogs support the latter. In addition, data catalogs are also an important tool to support the FAIR principles [Bor22; Lab20b; Qui20], data governance [Che22; Rya22; Sha16c], and data democratization [Eic22a; Lef21] by providing corresponding metadata and metadata management functionalities.

2.2.1 Historical Context of Data Catalogs

Although the focus on data catalogs has only been around for a few years, the idea of providing structured information about available data dates back to the 1960s. It becomes apparent that documentation technologies also become more sophisticated as the amount of data increases [Sen04b]. Figure 2.1 shows the evolution of this development. Initially, only indices for finding data on tape and simple field names were documented, but over time, additional information was added to better describe the data. Data catalogs represent the peak of current development in this regard. They store a multitude of metadata that should provide users with as much insight into the data as possible without them having to access and interpret it themselves. The term *data catalog* is not an invention of recent years but goes back at least 30 years, if not longer. One of the first mentions in the literature was by Johnson and Cribbs, who, as early as 1990, pointed out the need to use a data catalog as more and more data was being generated and the overview of it was being lost [Joh90]. With the advent of the World Wide Web (WWW), even more data was made available. Shannon et al. published a paper on this in 2005, in which a data catalog was intended to inventory the data available on the Internet [Sha05]. By the 2010s at the latest, ODPs (see Section 2.2.3) became a big topic. Numerous publications were published on topics such as requirements analysis [Mar11], state-of-art discovery [Att15], interoperability [Maa10a; Mar13], or (meta-)data quality [Kuč13; Rei22]. It was only around 2016 that a new wave of publications started to look at data catalogs in an enterprise context [Bug22; Eic22b; Jah23; Lab20a; Sha16b; Zai17].

2.2.2 Definition of Data Catalogs

For developing a design theory, we need to define the scope of the artifact for which it is created [Jon07]. We will establish this scope by defining *data catalogs*. First, we examine which definitions can be found in the literature. Second, we will derive our own definition, which will be used as the scope of our design theory.

One of the first mentions of data catalogs, which also includes a definition, comes from the field of computer science.

> "A catalog is an inventory of data resources, with the most basic information about each, such as source, name, location in source, size, creation date and

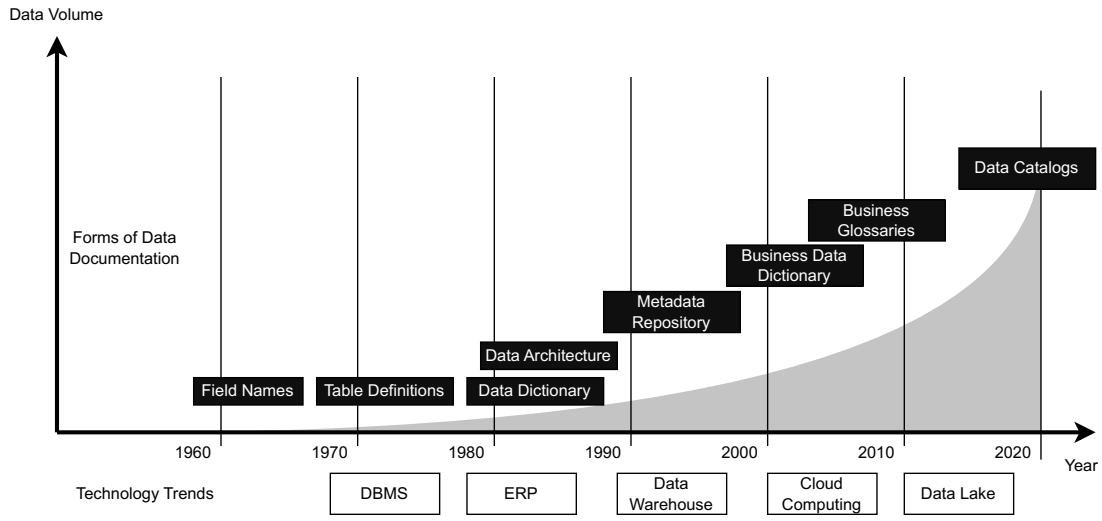


Figure 2.1: Evolution of data documentation [Kor19; Sen04b]

owner, and so forth."
— Franklin et al. [Fra05b, p. 29]

The definition considers data as a resource to be managed within a catalog. However, from today's point of view, this definition needs to be revised and can no longer be applied to modern data catalogs. In particular, the data stored in the catalog must meet modern data catalogs' increased requirements. It's no longer sufficient to store the most basic metadata. The data catalog must now contain as much additional context as possible, reflecting the depth and complexity of the modern data management landscape. In 2017, Zaidi et al. presented a new definition of data catalogs.

> "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."
> — Zaidi et al. [Zai17, p. 4]

For the first time, this definition clearly states that data is a commodity managed by companies from which value must be extracted. Furthermore, the stakeholders who work with and maintain the data play an essential role in this definition. Thus, the data catalog acts as an interface between data and users. Also, what was missing in the previous definition, that context information is crucial and should be stored in a data catalog, is mentioned here. In 2020, another definition was made by Quimbert et al.

> "Data catalogues provide information about data concerning one or many organizations, domains or communities. This information is described and synthesised through metadata records. Data catalogue centralised metadata is

> gathered in one location, usually accessible online through a dedicated interface."
> — Quimbert et al. [Qui20, p. 140]

In this definition, the term metadata is mentioned for the first time. It also addresses the importance of contextual information with "[...] information about data concerning one or many organizations [...]". Additionally, it refers to the fact that a data catalog is a central point of contact where users can obtain information about available data. While this definition introduces significant new concepts, it needs to follow up on one from previous definitions. Unfortunately, it no longer considers how a data catalog functions as a link between data and users. The latest definition of data catalogs was developed by Jahnke and Otto in 2023.

> "[Data Catalogs] are metadata management tools, that support the curation of data by providing capabilities to inventorize and discover data on an integrated platform, thus connecting data supply and demand."
> — Jahnke and Otto [Jah23, p. 90]

This definition again describes the concept that a data catalog mediates the supply and demand for data. For the first time, it is also evident that a data catalog is not just a simple repository for metadata but provides advanced functionality to inventory data and connect data and users.

The community still needs to reach a consensus on the definition of data catalogs. The core aspects of the definitions are: *data as an asset*, *storing relevant (context) metadata*, *matching data demand and data supply*, *provide additional functionality to meet objectives*, *stakeholders are all those who work with data*, and *inventory data*. Given the identified core aspects, it is imperative for this thesis and the scope of the design theory to propose a new, comprehensive definition of data catalogs.

> "Data catalogs are metadata management tools that inventory and curate data assets by storing relevant descriptive, administrative, and structural metadata about the data and provide sophisticated functions to help those working with data match the available data demand with the existing data supply."
> — own definition

2.2.3 Data Catalog Types

In the course of the work, various types of data catalogs will be mentioned. Also, our design theory strives to be as general as possible. It should apply to all types of data catalogs that fit the definition of data catalogs given in Section 2.2.2. In the following, we will provide a comprehensive overview of the existing types of data catalogs.

Data catalogs are used in many contexts and, depending on the application area, also focus on different goals. Depending on the context, a distinction is made between types of data catalogs in order to better describe which goals they focus on and in which scope they are used. Also, "data catalog implementations can vary depending on the use-case requirements, tool capabilities and maturity of each organization." [Zai17, p. 8]. To our knowledge, only the work of Jahnke and Otto exists so far, which attempts to identify the different types of data catalogs [Jah23]. They were able to identify seven types of data

catalogs. These are briefly described below. Table 2.1 also provides an overview of the types and their distinguishing core properties.

Table 2.1: Typology of data catalogs [Jah23]

Types	Organizational area	Integration	Metadata Management Scope	Data Management Level	Provider - Consumer Relationship
Enterprise Data Catalog	Intra-organizational	Stand-alone	Holistic	Metadata	Many-to-many
Context-specific Data Catalog	Intra-organizational	Stand-alone	Specific	Metadata	Many-to-many
Enterprise Data Management Platform	Intra-organizational	Module	Holistic	Data and Metadata	Many-to-many
Enterprise Data Marketplace	Intra-organizational	Module	Holistic	Metadata	Many-to-many
Data Spaces Data Catalog	Inter-organizational	Stand-alone	Holistic	Metadata	Many-to-many
Data Portal	Inter-organizational	Module	Holistic	Data and Metadata	One-to-many
Ecosystem Data Marketplace	Inter-organizational	Module	Specific	Both options possible	Many-to-many

Enterprise data catalogs are, as the name suggests, used in companies to increase the transparency of data and to support data democratization. In the catalog, the company's data is inventoried by only storing metadata, making it findable and accessible to the employees. The data is collected from the company's various data-holding systems. One of the significant challenges in data management is the integration of data sources into the catalog. This is particularly complex as the systems are often in different areas of responsibility. This type of data catalog must be able to map the data governance rules that prevail in the company. For example, responsibilities, information on terms of use, or legal restrictions on the use of the data should be possible to store. An enterprise data catalog is often described as a stand-alone software artifact that can be deployed independently of other software or platforms. An enterprise data catalog serves as a central point where all information converges, providing a comprehensive view of the company's data landscape. Employees can inventory and also search for data.

Context-specific data catalogs are only used in a narrowly defined context. Here, a further distinction can be made between technical and domain contexts. An example

of technical context is represented by the data catalogs Dataplex[1] from Google or Data Catalog[2] from Microsoft. These have been implemented specifically for inventorizing the data in the cloud storage of, for example, Amazon Web Services (AWS), Google Cloud, or Azure and can only be used with the respective provider. An example of domain context is the Science Mission Directorate (SMD) data catalog [Bug22], which is only used for data related to National Aeronautics and Space Administration (NASA) programs. Otherwise, this type of data catalog is similar to enterprise data catalogs.

Enterprise data management platforms are similar to Enterprise Data Catalogs. However, they differ in two essential points. First, they are not operated as stand-alone software but are always part of a more extensive software suite. When integrated with other modules, such as those for data governance or data quality, they significantly enhance the capabilities of existing data management systems. Accordingly, these data catalogs can often perform direct data management and access.

Enterprise data marketplaces build on the foundations of data catalogs. While usual data catalogs consider the data as an asset, in a data marketplace, it becomes the product [Eic22c]. Like in data catalogs, metadata about the data is stored and can be searched by interested buyers to find suitable offers. Data is offered on the seller's terms and can be purchased, similar to an online store for physical or digital goods. Data marketplaces can either fill specific niches or cover a broad range of offerings, as well as be public or only accessible to employees of a particular company [Azc22]. However, data marketplaces designed for use in a company have yet to be identified in practice [Jah23] and also face the question of what the motivation for offering the data is [Eic22c]. Eichler et al. nevertheless conclude that data marketplaces can support data democratization in enterprises [Eic22a]. Fernandez et al. also see enterprise data trading as an opportunity to get data out of its silos, as the data providers can be incentivized with some reward [Fer20]. Like enterprise data management platforms, enterprise data marketplaces are part of a more extensive software suite and are not operated as stand-alone software. In contrast, the data marketplace stores only metadata, and the actual data is stored elsewhere in the data provider's systems.

Data space data catalogs are a distinct type of data catalog optimized to support inter-organizational data exchange within the scope of data spaces. They contain particularly useful metadata for this purpose, such as usage conditions for the data. Thus, participants in data spaces can search for data offered by companies, identify suitable data for their projects, and make data offers themselves. The data catalogs are often a necessary part of a more extensive software solution that enables the connection to the data space. In contrast to data marketplaces, these data catalogs are based on the idea of data sovereignty and federal architecture. Each participant has its own data catalog, and there is no central place where the data and metadata must be collected.

Data portals, or ODPs, are often used in open data to make public data from states, cities, and municipalities available to the population [Att15]. Similarly, universities or

1 `https://cloud.google.com/dataplex`
2 `https://azure.microsoft.com/en-us/products/data-catalog`

institutions such as NASA use data portals to publish data from their projects and research programs [Bug22; Tzi21]. The data in the data portal is made available to individuals and companies. The portal usually provides direct access to the data by providing a download link. The data portal is filled with metadata either manually or via data harvesting. With data harvesting, ODPs collect relevant entries from other portals and offer them themselves. The data catalog is regarded as one of many modules (such as data visualization) within the portal.

Ecosystem data marketplaces are data catalogs that enable companies to buy and sell data. The difference to enterprise data marketplaces is that data is traded not only within a company but also between different companies through this system as a data trustee. The data catalog here is part of a larger system, which, for example, also contains typical store functionalities such as shopping cart or payment.

2.2.4 Data Catalog Solutions Relevant to the Thesis

There are already numerous data catalog software solutions available for various applications. The following provides an outline of the data catalog solution we will use in this work. We have decided to only look at Free and Open-Source Software (FOSS) solutions, as one can already gain insights without purchasing the software.

DataHub was developed by LinkedIn[1] and is available under the Apache-2.0 license. The software architecture can be divided into three areas. The *App Tier* contains the logic of the software and provides services to perform metadata management and operate front-end applications. Metadata management can be implemented on a federal basis. This allows different departments to own metadata and make sovereign decisions about the sharing and use of metadata. The *Persistence Tier* includes all databases used in DataHub. Metadata is stored in MySQL[2] and in Elasticsearch[3]. MySQL is used as the main storage, while Elasticsearch has the specific task of handling user search queries. A Kafka[4] instance is used to log changes made to the metadata. All services that connect to the user's system and collect metadata run in the *Client Tier*. DataHub already comes with a wide variety of connectors[5] to integrate with various technical systems. DataHub is still used internally at LinkedIn.

OpenMetadata is also available under the Apache 2.0 license and is mainly developed by Collate[6]. The software has a similar architectural approach to DataHub. On the one hand, there are several databases in which metadata is stored. MySQL and Elasticsearch are also used here in the same way. There is a kind of *App Tier*, where the logic of the software is located, and things like authentication, event handling, and metadata management are handled. In addition, there are also connectors[7] to integrate with common data-holding systems.

1 `https://www.linkedin.com/` [Accessed: May 10, 2024]
2 `https://www.mysql.com` [Accessed: May 10, 2024]
3 `https://www.elastic.co/` [Accessed: May 10, 2024]
4 `https://kafka.apache.org/` [Accessed: May 10, 2024]
5 `https://datahubproject.io/docs/metadata-ingestion/source_overview` [Accessed: May 10, 2024]
6 `https://www.getcollate.io/` [Accessed: May 10, 2024]
7 `https://docs.open-metadata.org/v1.3.x/connectors` [Accessed: May 10, 2024]

Comprehensive Knowledge Archive Network (CKAN) is an open source data portal available under the AGPL-3.0 license. In contrast to the other data catalogs, the source code of CKAN must be published if the code is modified or extended. CKAN is often used to create data portals for open data projects. Over the years, several standards and plugins have been developed for CKAN, enabling its use in many contexts. For example, standardized metadata models can be used, and harvesting from other ODPs is supported. The EU data portal[1] is one of the best-known ODPs based on CKAN [Kir19b]. The EU data portal is the central point of contact for all data from the EU and its member states.

Finally, we will look at the **Eclipse Dataspace Components (EDC)[2] Federated Catalog**. The EDC make it possible, among other things, to build a federated data catalog for data spaces. The data catalog is usually deployed with a data space connector and contains the data that can be shared via a data space. The owners of the data catalog can also store more specific metadata, such as usage policies in the Open Digital Rights Language (ODRL) so that other participants in the data space know how they are allowed to use the data. The data catalog follows a federated approach that allows data space participants to retain control over their metadata and decide which metadata they want to share and with whom.

2.2.5 Additional Terminology Related to Data Catalogs

During the thesis, some terms related to data catalogs are used regularly. These terms will be briefly defined below.

In the scope of this thesis, a **data network** is a graph whose nodes mainly represent the data inventoried in a data catalog and whose edges represent the relationships between the data. If the data catalog also inventories entities other than data, such as services, apps, publishers, or products, these can appear as nodes in the data network. In this case, the nodes may be labeled. The nodes can also contain any other metadata about the entity, such as an ID or a title. The edges of the network can represent any relationships. Again, it is helpful to label the edges to describe the relationship between the nodes. Furthermore, in most cases, it is necessary to use directed edges. Also, further payloads can be attached to the edges. For example, scores that describe the similarity between two inventoried data sets.

Data profiling is often used to describe the metadata discovery process. There are many views and definitions of this term in the literature. Johnson describes data profiling, for example, as something that "[...] refers to the activity of creating small but informative summaries [...]" [Joh09]. A somewhat broader definition comes from Abedjan et al.: "Data profiling is the set of activities and processes to determine the metadata about a given dataset." [Abe15, p. 557]. Often, data profiling is associated with extracting the schema of data, identifying data types in columns of tabular data, checking for null values, or collecting other relevant data quality information. However, in data catalogs, we use this term to refer to all kinds of metadata extraction and calculation. This ranges from the

1 `https://data.europa.eu/en` [Accessed: May 10, 2024]
2 `https://github.com/eclipse-edc/FederatedCatalog` [Accessed: May 10, 2024]

simplest tasks, such as determining a file's size, to more complex tasks, such as content analysis.

CHAPTER 3

Related Work

We conducted an Systematic Literature Review (SLR) and generated a data catalog corpus to identify related work to our thesis. First, we will describe how we generated the corpus. We then summarize the research literature we identified. There, we also discuss existing DPs for data catalogs.

3.1 Generating the Data Catalog Corpus

The literature corpus was generated according to a structured procedure based on Webster and Watson [Web02]. In the **first phase**, an initial corpus of articles was collected. For this, the meta bibliography Scopus[1] was used, which indexes several other bibliographies and thus offers an extensive peer-reviewed literature base for searching. The search string used was:

> 'TITLE-ABS-KEY("Data Catalog*") AND (LIMIT-TO (SUBJAREA, "COMP")) AND (LIMIT-TO(DOCTYPE, "cp") OR LIMIT-TO(DOCTYPE, "ar")) AND (LIMIT-TO(LANGUAGE, "English"))'

By narrowing down to *Data Catalog**, all articles dealing with data catalogs and cataloging are to be selected. Furthermore, the articles are narrowed down to publications in the field of computer science that were published as conference papers or journal articles and are in English. This resulted in 266 potential articles. We then removed duplicates, non-English, and inaccessible articles. Finally, abstracts were read from all remaining articles. Reading was done to identify whether the article's core dealt with data catalogs or strongly related topics such as metadata management or vocabularies for data catalogs. After completion, 78 relevant articles could be identified, which represented the initial corpus. A backward search was conducted in the **second phase**. Each of the 78 papers was reviewed, and irrelevant articles were excluded based on their titles. The abstracts of the remaining papers were read, identifying 21 relevant articles. In the **third and final phase**, a forward search was conducted to identify papers citing those already in the corpus. Irrelevant articles were excluded based on titles, and abstracts of the remaining articles were read, resulting in 13 additional relevant articles. This brings the total corpus to 112 articles. The complete corpus can be found in the Data Catalog Corpus chapter.

1 `https://www.scopus.com`

3.2 Discussion of the Data Catalog Corpus

The corpus contains many works that contribute design knowledge to the knowledge base of data catalogs. In the following, we refer to contributions in the form of artifacts as design knowledge, even if they were not developed as part of a DSR project. First, it can be noted that many papers report on the development or operation of **context-specific data catalogs** [Fri14; Han13; Oli16b; Pes15; Sha05; Tim23; Won23]. One example from the corpus is the work on a data catalog for geothermal exploration [Cor10]. The authors identify problems with the existing solutions for searching geothermal data and propose a solution based on a data catalog. Existing data model standards from the geothermal sector are discussed, and a new data model for the data catalog is proposed. Furthermore, topics and keywords that can be used to tag the data are identified, and a prototype of the data catalog is presented. Most of the works are designed as described in this example. In addition to explicit knowledge, these works contain a lot of implicit design knowledge implemented in the data catalogs and cannot simply be reused. For example, a screenshot of the application is demonstrated in the work of Corbel and Poulet [Cor10]. The screenshot most likely contains design knowledge about how the application should be designed for the geothermal domain. However, it is not made explicit.

Several works were identified that report on design knowledge in the form of **architectures**. A more general work is that of Sen [Sen04a]. There, the effort to develop "a design methodology that can capture the metadata scattered in the data warehouse projects and put them in the metadata warehouse." [Sen04a, p. 171] is described, and three architectural styles are presented to solve the challenges. More concrete architectural proposals come from Halevy et al., where they describe how Google deals with the mass of data and metadata [Hal16] and from Bugbee et al., who discuss which database architecture is most promising for the NASA Science Mission Directorate data catalog [Bug22]. Further work is in the area of CKAN. Scholz et al. describe how CKAN can be extended by Hadoop to store the data itself in addition to the metadata [Sch17]. Kirstein et al. report on the architecture and components of the EU ODP [Kir19a], as well as an architectural proposal on how to better persist unique identifiers of datasets [Kir23].

Other works contribute design knowledge in the form of **methods**. Many of these methods are aimed directly or indirectly at improving data discovery, for example, by using ontologies [Lee12], tagging data [Jia19; Tyg16], or describing recommender engines [Fen17]. Many studies are also concerned with determining the similarity between data and metadata, thus suggesting better search results [Ber22; Sen18; Ško19; Sko19]. The use of knowledge graphs to improve the search functionality of data catalogs is also discussed in several papers [Neu18a; Neu18b; Ojo20]. Methods for measuring the usability of the data catalog and the data inventoried there [Dor21; Gon07; Has20; Kub16; Neu16] and also the measurement and improvement of the quality of the metadata [Aik20; Kub18; Kuč13; Nog21; Urb22] are discussed. Unique methods for visualizing data in data catalogs should help users to better understand the data [Ben14; Car15].

Research has also developed data **models** for various domains, including data governance [Joh90; Kře19; Maa10a; Rya22] and specific sectors like water information [Dih15] and transportation [Scr22], as well as chatbot enhancements [Cap17]. Additionally, vocab-

ularies and ontologies are applied to Application Programming Interfaces (APIs) for better interoperability [Ade17].

More general design knowledge was developed by Ehrlinger et al. as part of an SLR [Ehr21a]. They were able to identify four components that are relevant to the implementation of data catalogs. **Metadata Management** is seen as crucial for data catalogs, focusing on data descriptions, like quality and access control [Sha16b]. Modern strategies should adopt metadata standards and ontologies, like DCAT [Sko19]. **Business Context** should be considered, as the data catalog's primary consumers are business users rather than IT specialists [Lab20a]. This can be accomplished by either extending metadata with business context attributes [Sha16b] or creating a company-wide business glossary [Lab20b], which serves as a central library of agreed-upon business words. Data visualizations in the data catalog can also help to better understand and use the data [Car15]. **Data Responsibility Roles** should be defined in the context of a data governance framework [Lab20a], as poor data quality in companies is mainly because there are no clear responsibilities for data. Finally, the **FAIR principles** were identified as an essential component for data catalogs. Data catalogs can contribute to making data FAIR by improving the findability, accessibility, interoperability and re-usability of data [Lab20a]. In particular, this can be done through mature standardization [Bor22].

Literature also discusses how data catalogs can support data democratization [Eic22a]. "[Data democratization is] an ongoing process of enabling digital data access to both technical and non-technical users to understand, find, access, use, interact and share appropriate data within the boundaries of legal, confidentiality and security limitations by transferring data ownership and responsibility to empower users for efficient and accurate decision-making, promote collaboration, and create a knowledge-sharing culture in an organization." [Sam23, p. 3]. Lefebvre et al. found five enablers that support data democratization, where two of them can be supported by data catalogs [Lef21]. **Collaboration and knowledge sharing** is one thing that data catalogs are primarily built for. For example, metadata stored in data catalogs can be added or corrected by different users. Solutions such as review systems can also be used in data catalogs to share further knowledge, e.g., about the data's successful or unsuccessful application. **Broader data access** is also addressed by data catalogs. On the one hand, the data catalog makes it possible to discover data. Furthermore, in well-maintained data catalogs, information on how to access data is stored. In addition to technical access, this also includes information on who may access this data and where it may be used. Ideally, contact persons who can support the user in accessing the data are stored.

Studies also report on typical **challenges** that can prevent the successful implementation of data catalogs. Portisch et al. see a lack of a consistently used base ontology, metadata quality and multilingualism as typical problems in data catalogs [Por20]. Nikiforova and McBride have analyzed several ODPs and found that they usually have weaknesses in the areas of machine-readable formats, documentation and tutorials, social media and sharing, visualization and statistics, and user rating and comments [Nik21]. Won et al. looked at CKAN in particular and identified five challenges: Data management limitation, no real-time feature, lack of metadata, no interconnection standard, and low utilization [Won23].

Only the work by Zuiderwijk et al. follows an DSR approach and reports on **Design**

Principles for making Open Government Data (OGD) easier and faster to use [Zui16]. It is about how a system, which in our understanding can be called an ODP, has to be designed. Zuiderwijk et al. identified three principles: *metadata*, *interaction mechanisms*, and *quality indicators*. The principle of *metadata* states that the system should strongly focus on metadata. In concrete terms, this means that, for example, metadata should be used in the search function, metadata should be able to be loaded from other ODPs, insights into the data should be made possible without downloading the data, or, for example, that users should be able to maintain metadata. The principle of *interaction mechanisms* states that the users of the ODP should be able to participate passively and actively. They can, for example, participate in discussion forums, write wiki articles, or make data available on the portal. The principle of *quality indicators* states that the ODP should have mechanisms to assess the data. Specifically, this can be done by assessing the quality of the data in various quality dimensions, by making a review system available, or by providing information about the person who assessed the quality of the data.

The related work suggests that further research is needed on practice-based and design-oriented knowledge about data catalogs. Several reasons support this decision. First, the design knowledge found in the work of Zuiderwijk et al. is not formulated in a standardized format, nor is it prescriptive. Although the authors give concrete examples of the implementation of their DPs, which certainly have the character of DFs, it is difficult to grasp the actual essence of the DPs from their work. Especially in the case of the *metadata* principle, the question arises as to what exactly is the mechanism for implementation. Furthermore, the work focuses on ODPs and may not be applicable to other types of data catalogs. Also, the design knowledge is not grounded in practice. It was prototypically tested but did not emerge from practice. This makes it challenging to make the knowledge applicable to practitioners. Second, most corpus articles delve deeply into certain aspects of data catalogs. None holistically examines data catalogs and provides practice-based, design-oriented guidance for creating successful ones. Therefore, we still consider the field of data catalogs incomplete in terms of prescriptive design knowledge and design theory.

CHAPTER 4

Research Design

To respond to our MRQ, we developed a design theory for data catalogs. In the following, we describe our research methodology and how we developed the components of the design theory according to Jones and Gregor [Jon07]. We will first give an overview of our DSR project, followed by how we developed our design theory for data catalogs.

4.1 Overview of the DSR Project

In order to maintain scientific rigor, the development of the design theory was planned and conducted as a DSR project. DSR is a research methodology that emerged from the field of business informatics and information systems [Öst10] and is now also used widely in software engineering [Bar22a; Woh21]. "Whereas natural science tries to understand reality, design science attempts to create things that serve human purposes. It is technology-oriented. [...] Rather than producing general theoretical knowledge, design scientists produce and apply knowledge of tasks or situations in order to create effective artifacts." [Mar95, p. 253]. The main emphasis is on developing and improving theories or artifacts iteratively while preserving relevance and scientific rigor [Hev04]. The term artifact encompasses a variety of things. Vaishnavi et al., for example, include *constructs*, *models*, *frameworks*, *architectures*, *design principles*, *methods*, *instantiations*, and *design theories* [Vai04]. Generally speaking, an artifact "[...] is something created by people for some practical purpose" [Wie14, p. 29]. Due to its practical relevance, DSR supports the demand for industry and academia collaboration, which is considered very important [Run14].

While there exist many works that provide guidelines for conducting DSR [Pef08; Sei11], we decided to use the work of Vaishnavi and Kuechler [Vai04], as it represents the best framework according to Venable et al. "[i]f you want to develop design theory" [Ven17, p. 9]. We conducted a single DSR iteration. Figure 4.1 illustrates our approach, which we will briefly discuss.

Problem Awareness: At the beginning of the DSR process, identifying the problem to be solved is paramount. For example, this identification can be achieved by deriving and adapting current industry trends to one's research area. Studying research-specific or external literature is also a valid methodology for identification. Problems can also be brought to one's attention by practitioners. As previously discussed, data catalogs represent a significant technical solution for supporting FAIR principles, data governance, and data democratization. We identified a lack of research focusing on data catalogs regarding holistic, practice-based, and design-oriented knowledge (see Chapter 3). We did so by conducting an SLR according to Webster and Watson [Web02].

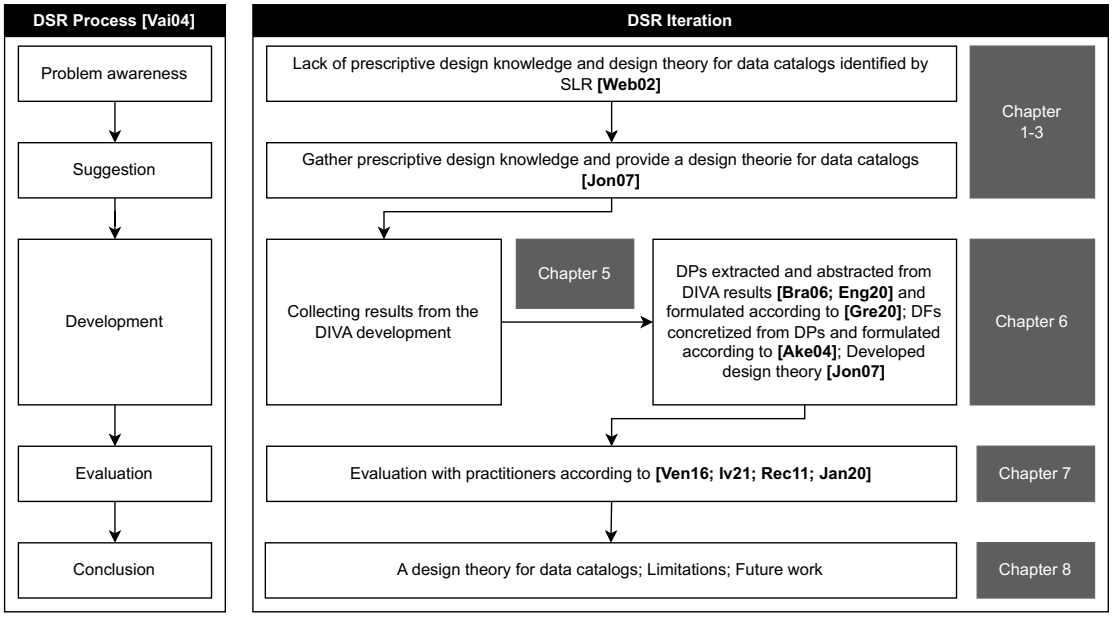


Figure 4.1: Overview of our DSR project

Sugession: In this phase, a solution proposal is developed that should address the problem identified in the first phase. We decided to develop a design theory [Jon07] for data catalogs that supports researchers and practitioners in designing successful data catalogs. The design theory should give a holistic view of the main aspects of what a data catalog must do. To ensure practical grounding, the design theory was to be based on an existing data catalog that had been established and used for several years.

Development: In the development phase, the artifact that is supposed to contribute to solving the problem is created or improved. We developed a design theory for data catalogs mainly by extracting DPs and DFs from our self-developed data catalog Data Inventory and Valuation Approach (DIVA). To do this, we first collected the results we developed during the development of the DIVA data catalog. These can be found in Chapter 5. We followed Engström et al. [Eng20] and abstracted our results from the solution instance into DPs. The DFs were generated by concretizing the DPs, whereby only those DF were taken that are grounded in DIVA to ensure practical relevance. Finally, the design theory was compiled, and all missing components were developed. The details of our development process can be found in Section 4.2. The results can be found in Chapter 6.

Evaluation: The evaluation aims to determine whether the artifact solves a problem and how effectively it works. Depending on the type of artifact, the evaluation methodology differs. We focused mainly on evaluating the DPs, as they are the most important component for a design theory [Hei14]. We conducted a formative and artificial evaluation [Ven16]. There, we tested the reusability according to Iivari et al. [Iiv21], and the ontological expressiveness according to Recker et al. [Rec11] and Janiesch et al. [Jan20]. Details on the evaluation design and the results can be found in Chapter 7.

Conclusion: In this process step, the research results were critically examined. Here, the successes, as well as the limitations, were highlighted. Making the results publicly available to the communities of scientists and practitioners is also essential. We are doing this by publishing this work. The conclusion can be found in Chapter 8.

4.2 Development of the Design Theory

A design theory requires the provision of eight specific components according to Jones and Gregor [Jon07]. (1) *Purpose and Scope* describes the type of artifact for which design theory can be applied and where the limits lie. (2) *Constructs* refer to important entities necessary for understanding the theory. In other words, it is also the core vocabulary of the theory. (3) *Principles of Form and Function* can be associated and populated with DPs. It should contain knowledge in the form of blueprints describing which situation one must apply which mechanism to achieve a particular effect. (4) *Artifact Mutability* describes what kind of changes can be endured by the artifact that implements the design. (5) *Testable Propositions* are constructed as truth statements and can be checked against the data catalog that implements our design. They can be used to check whether the design theory has been successfully instantiated. (6) *Justificatory Knowledge* describes the foundation on which the design theory is established. Fundamental works that represent kernel theories and thus support the design theory and informal knowledge can serve as a basis here [Gre13a]. (7) *Principles of Implementation* contains knowledge about transforming the principles of form and function into a tangible artifact. This component can best be described as a collection of DFs. It is intended to bridge the gap between abstract knowledge and the actual implementation. (8) *Expository Instantiation* can be, for example, a software artifact that implements the principles of form and function. Ideally, the design theory is accompanied by an instantiation that can be used for demonstration or testing.

In the following, we describe how we developed the components *Principles of Form and Function* and *Principles of Implementation*. The development and results of the other components emerge naturally in the course of the work. As a result, we will not go into greater detail about them here.

Principles of Form and Function

When developing DPs, two different perspectives can be adopted [Möl20]. The *supportive* perspective is an ex-ante approach in which DPs are developed before the actual design process to support the actual artifact's development. In contrast, the *reflective* perspective is an ex-post approach, where DPs are developed after the design process. There, "[...] reflecting upon what has been done is required [...] and DPs need to be abstracted." [Gre09, p. 7]. We decided to follow the reflective approach and extract both the DPs for answering RQ1 and the DFs for answering RQ2 from a data catalog software artifact. This is done to ensure the practical grounding of our design knowledge.

As a software artifact, we decided on the DIVA data catalog. DIVA is a data catalog developed at the Fraunhofer Institute for Software and Systems Engineering ISST (Fraunhofer ISST) between the year 2017 and 2023. The software development process of DIVA was mainly led by the author of this thesis. The author of this thesis and changing

student assistants at Fraunhofer ISST were involved in the actual implementation of DIVA. Collecting problem awareness, developing solution proposals, and evaluating the results were done in close collaboration with industry partners. With DIVA, we have a design process for a data catalog that we supervised, which can be used as a basis for extracting design knowledge [Möl20]. This guarantees that our design theory impacts practice and research [Bas18a]. In Chapter 5, we detail how we developed DIVA and what results we have achieved.

Since DIVA is a software artifact, the extraction of DPs was fundamentally guided by the work of Engström et al. [Eng20]. Based on this, DPs can be achieved by abstraction of the solution instance. Abstraction "refers to the activity of identifying the key design decisions for a defined scope of validity of a solution" [Run20, p. 131]. The DPs established in this thesis are intended to be applied to the entire class of data catalog systems, with a broad scope of application that is not limited to specific domains. They should, therefore, be equally valid for all data catalog types mentioned in Section 2.2.3. To increase their comprehensibility, complexity was reduced to simplify implementation in concrete instantiations.

Wache et al. identified two dimensions for adjusting the formulation of DPs [Wac22]. The first dimension is the level of abstraction, where more abstract DPs have broader applicability but may lose important details for concrete instantiations. The second dimension concerns the density of concepts in a DP, including various users, developers, goals, contexts, mechanisms, and the like. Based on the requirements mentioned above, the DPs are positioned in Quadrant III of the positioning framework according to Wache et al. [Wac22] (see *DPs* in Figure 4.2). In contrast to DIVA, abstraction should be maximized and concept density minimized. Accordingly, DIVA is located in Quadrant II with the lowest abstraction and a high degree of concepts (see *DIVAs* in Figure 4.2).

Braun and Clarke's work on *thematic analysis* shaped our process of abstraction [Bra06]. It is intended to identify patterns in data, making it ideal to identify overarching themes for our DPs. The process is divided into six phases. (1) Familiarizing ourselves with the data was done by revisiting the DIVA software. We have reviewed all development iterations and analyzed and documented both context and software in detail (to be found in Chapter 5). (2) We produced a coding of the noteworthy DIVA development outcomes (e.g., Data Assessment Capability (R1.1)). These codings are referred to as DIVA development results. (3) We used these results and tried to group them and assign them to more abstract themes. (4/5) We performed the review and refining phase iteratively. The iterations led to further abstraction or concretization, generation, removal, summarization, or partitioning of DPs. We have investigated whether each DP is found within DIVA. The procedure was performed until the set of DPs stabilized. (6) We have finalized the results and incorporated them into the context of this thesis.

We then supported our set of DPs with existing literature. For this purpose, we reused the corpus generated in Chapter 3. The abstracts of the articles were reread and related to the DPs in terms of content. This approach seems appropriate because the abstracts should already contain the work's main challenges, goals, methods, and results. If, for example, an article elaborates a solution in the context of data catalogs or strongly related topics, which directly pays into implementing a DP, the article was assigned to the corresponding

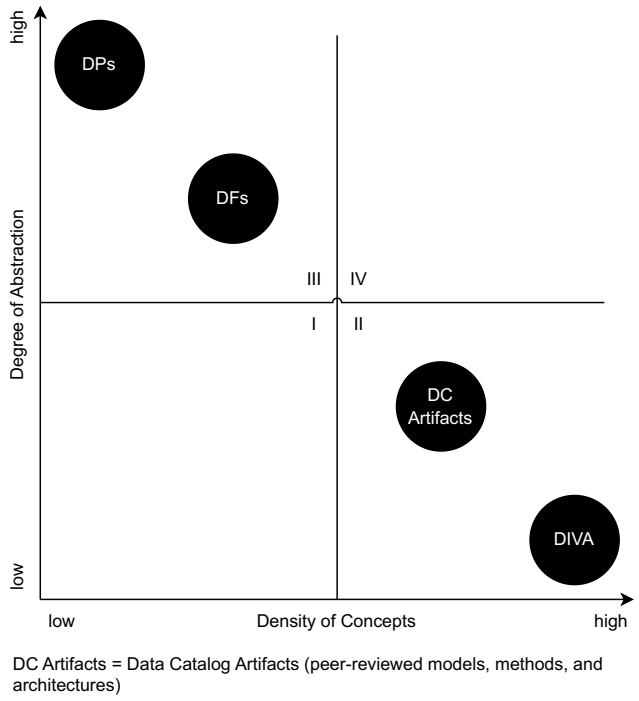


Figure 4.2: Classification of the design knowledge for data catalogs using the positioning framework of Wache et al. [Wac22]

DP.

In literature, several works have established templates for formulating DPs [Cha15b; Cro18; Gre09; Gre20]. We decided to follow the latest approach of Gregor et al. [Gre20]. This work develops a set of additional knowledge for the DPs. First, the DPs in this thesis were extracted from DIVA, so it should be noted where the DPs originated from. Second, DFs were also developed to assist the target group of the DPs in implementing them. Third, we identified supporting literature for the DPs. For this reason, we extend the tabular structure of Gregor et al. to include *Extracted from DR (DIVA Result)*, *Supported by L (Literature)*, and *Realized by DF (Design Feature)*. Since the DPs apply to all classes of data catalogs and any domain, we omit the context specification. The formalization of the DPs adapted to our needs can be found in Table 4.1.

Principles of Implementation

After finishing the *Principles of Form and Function* component, we developed a set of DFs that populate the *Principles of Implementation* component. We did this by concretizing the DPs. This means that the DFs contain more concepts that must be understood during implementation. In our case, these are often details about existing technologies embedded in the DFs. The DFs can still be positioned in Quadrant III of the positioning framework (see *DFs* in Figure 4.2). The DFs were limited by what has been developed in DIVA over the years. Therefore, no DFs were added that are not found in DIVA, ensuring practical

Table 4.1: Components of the DP schema based on Gregor et al. [Gre20] and adapted to own needs

Design principle title	
Structure	**Components***
For Implementer I to achieve or allow Aim A for User U	Aim, implementer, and user
Employ Mechanisms M1, M2, M3 Involving Enactors E1, E2, E3	Mechanisms: (acts, activities, processes, form/shape/architecture, manipulation of other artifacts) Subsidiary components/artifacts can have their own DPs
Because of Rationale R	Rationale: Theoretical or empirical justification for the DP
Extracted from DR (own extension)	Extracted from: List of DAIs results that contributed to the extraction of the DP
Supported by L (own extension)	Supported by: List of publications that support the DP
Realized by DF (own extension)	Design Features: List of DFs that help in the practical implementation of the DPs

* Note: In many explications of DPs, some components are not made explicit

relevance.

The thesis also includes our peer-reviewed articles with design knowledge applicable to implementing the DFs. The design knowledge in the published articles is even more concrete than the DFs. They, therefore, also contain more concepts. Their positioning in the positioning framework is in Quadrant II (see *DC Artifacts* in Figure 4.2). Which publication enriches which DF with design knowledge is discussed accordingly during the presentation of the results.

We take a more straightforward approach when formulating the DFs. We do not need to present as many concepts and other elements here. Therefore, we decided to formulate the DFs according to the template of van Aken [Ake04]. In the further elaboration of the DFs, details on the rationale, implementation recommendation, and application examples will be given.

CHAPTER 5

DIVA Development

DIVA was developed iteratively over more than six years and forms the essential basis for the extraction of DPs and DFs for our design theory. We describe the development structured in eight iterations. DIVA was not explicitly developed as part of a DSR research project. Nevertheless, we use the structure of the DSR framework by Vaishnavi and Kuechler [Vai04] *retrospectively* to describe the development of the iterations and give it a structure. During this thesis, information concerning industry partners was anonymized due to missing permission to disclose the information. The individual iterations are described from memory, as we do not want to disclose any information on partners accidentally. Some results were extracted as peer-reviewed publications and document the results more rigorously. If this is the case, we will explicitly mention it. DIVA is available as FOSS at GitHub.[1] The results of the iterations are presented and numbered (e.g., R1.1). The enumeration is necessary so that they can later be referred to again by us when assigning them to the DPs. We chose the results based on the focus of each development iteration.

5.1 Iteration 1

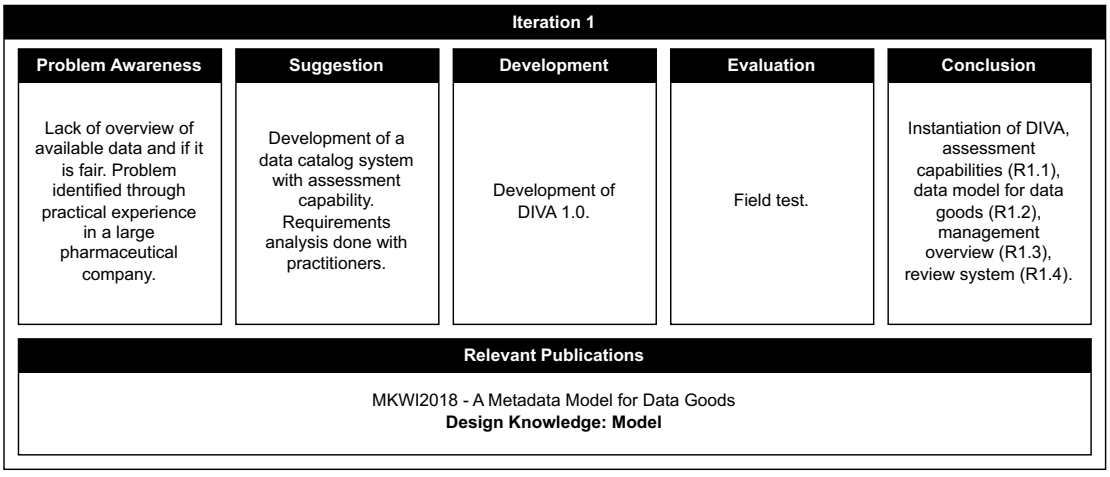


Figure 5.1: Summary of the DIVA development iteration 1

In 2017, we initiated the DIVA development in collaboration with a large German pharmaceutical company. The company generates or purchases data in many areas,

1 `https://github.com/FraunhoferISST/diva` [Accessed: May 24, 2024]

whether in research or production. Here, they struggled with two significant challenges. The first is the amount of data partitioned within the company. Often, employees did not know what data is available in-house. In the worst case, data was re-collected or expensively purchased a second time, resulting in duplications. The second challenge concerns the value of the data. Often, data was purchased at a high cost. Also, employees decided whether to use and buy data based on gut feeling. If there were ways to value the data better, assessing whether the data's cost is justified would be possible. This project aimed to provide employees with an overview of existing data and indicators of whether data is *fair*. Fair means that the Return on Investment (RoI) from acquiring and using the data must be sufficient in this context. To address these challenges, a data catalog was developed to inventory the pharmaceutical company's data. Particular attention was paid to viewing the data as an asset and providing assessment capabilities to identify whether the data is fair.

Development

A working data catalog, called DIVA, was developed to meet the requirements of the pharmaceutical company. DIVA 1.0 was designed as a monolithic application. It is a classic web application with a client, backend, and database. In order to make the best use of our limited resources, established technologies were selected. Technologically, the data catalog is based on Node.js[1] with Keystone[2] as a Content Management System (CMS) in the backend, MongoDB[3] as the database and the Bootstrap[4] framework in the frontend for a consistent look and feel.

To store relevant metadata in DIVA that allows for an assessment of data, the Metadata Model for Data Goods (M4DG) was developed that describes data as an asset. This metadata model served as a schema for MongoDB, allowing it to store and search the metadata in the catalog. Further experts from industry and research were consulted during the development of the model. Workshops were held to develop specific data attributes that are relevant in practice and have not yet been reflected in metadata models for data catalogs. This includes, for example, the usage license, whether a service level agreement exists, how much the data costs, whether a discount is available, or who is liable. Because the model is based on DCAT, it is still interoperable, which means that metadata records can be exchanged with other data catalogs that support DCAT. We will mention this model again later when we describe details of our design theory because it contains concrete **design knowledge in form of a model** that practitioners can apply. Details regarding the M4DG can be found in our publication *A Metadata Model for Data Goods*, which is part of this thesis [Spi18, Paper I.].

Using DIVA 1.0, employees can search for data (see Figure A.1), view detailed information (see Figure A.2), inventory new data (see Figure A.3), get an overview of the data catalog content via a management dashboard (see Figure A.7), and to review the data (see top

1 `https://nodejs.org/` [Accessed: July 10, 2023]
2 `https://keystonejs.com/` [Accessed: July 10, 2023]
3 `https://www.mongodb.com/` [Accessed: July 10, 2023]
4 `https://getbootstrap.com/` [Accessed: July 10, 2023]

right of Figure A.9). During inventorizing, other important information can be provided, such as alternative data, providers, responsibilities (see Figure A.4), or price information about purchase costs, storage costs, or maintenance costs (see Figure A.5). Data can also be assessed by rating different quality dimensions (see Figure A.6). The data assessment is then presented to the user visually in the form of a radar chart (see Figure A.9). Thus, combining alternative data, user ratings, and costs, it is possible to determine whether the data is *fair*. Freely available data can be accessed directly via an *Access* button. If access is restricted, one can directly contact the responsible person (see *Access* and *Call Owner* button in Figure A.2).

Evaluation

The evaluation was performed as a field test in the context of the pharmaceutical company. We deployed DIVA on the infrastructure of the pharmaceutical company and made it available to several employees. These employees filled the catalog with many well-known data in the first step. Subsequently, the data catalog was piloted by employees for day-to-day use. They were generally satisfied with the solutions and could comprehend the added value of a data catalog. They could use the catalog and, when inventoried, find suitable data for themselves. However, this is where the most significant point of criticism arises. The initial filling of the data catalog and keeping it up to date was very time-consuming. Also, not all the pharmaceutical company's available data was inventoried in DIVA, which is why the catalog could only suggest meaningful results occasionally. The employees wished for a much more up-to-date and complete data catalog without the time-consuming manual maintenance.

Conclusion

Several results were achieved in this first development iteration. One is the DIVA data catalog software, which can be used to inventory data and prevent unwanted data duplication. We developed the assessment capability, which allows users of DIVA to decide better whether the data represents a fair offer (R1.1), the management dashboard that gives an overview of the data catalogs content (R1.3), and a review system (R1.4). The source code for DIVA 1.0 has not been published. Finally, the M4DG metadata model is to be mentioned, which describes data in terms of an economic good (R1.2). The data model was published in *A Metadata Model for Data Goods* at the Multikonferenz Wirtschaftsinformatik (MKWI) 2018 [Spi18, Paper I.]. Figure 5.1 summarizes this development iteration and knowledge contributions.

5.2 Iteration 2

The underlying project for the second development iteration was part of a research consortium with industry and research partners. The research consortium was mainly concerned with securely and confidently exchanging data between companies. An important aspect of the project was the use of data catalogs. Companies can publish their own and view other companies' data offerings via these. As in the previous iteration, selecting suitable data for one's project was challenging. The question to be answered was how to help companies evaluate the data offerings and thus identify suitable data. The basic

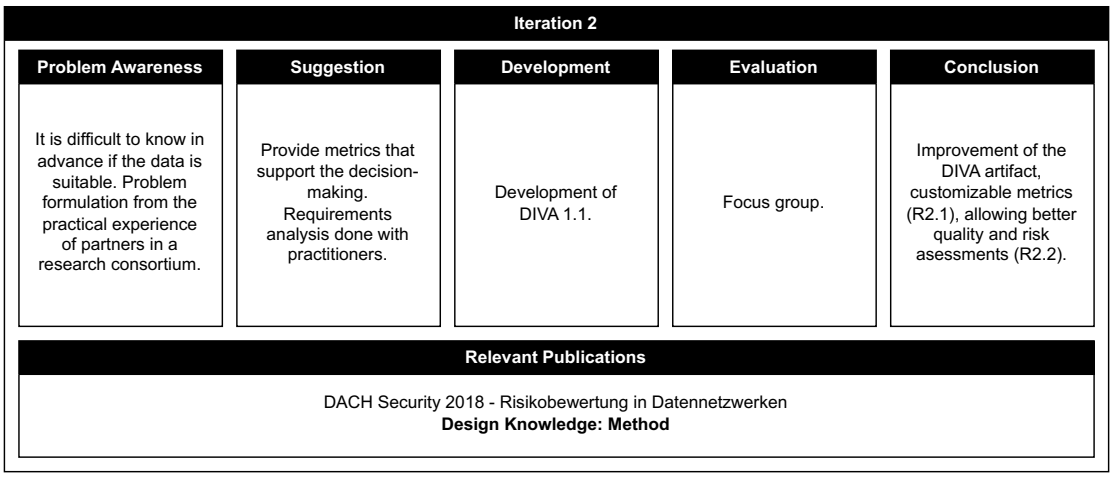


Figure 5.2: Summary of the DIVA development iteration 2

idea was to store evaluation metrics in the data catalog. Based on the metadata of the inventoried data, these metrics can provide users with an indicator of whether the data is suitable for their project. At the beginning of the project, the industry partners expressed many ideas about which metrics could be helpful. For this reason, the possibility of users adding metrics at a later stage was considered.

Development

DIVA has been extended with the possibility to display and add one's own metrics. Experts can add metrics via the user interface (see Figure A.8). Through a code editor embedded in the website, users can access metadata and thus calculate their own metrics. JavaScript serves as the programming language here, as it can be executed directly in the browser. The execution itself takes place in a sandboxed iFrame to prevent endless or long-running code from negatively affecting the data catalog's performance. Furthermore, only read actions on metadata are allowed. A metric consists of several parts. A metric has a title and a description so that users can understand its target. Additionally, a metric can be turned off if it is found to be causing problems. Metrics can also be assigned to groups, only displayed for certain types of inventoried data. Users can then view the metrics on the web interface (see Figure A.9).

Evaluation

We presented our findings to a focus group of research consortium members. One focus group participant suggested using the metrics to measure the degree of completeness of the metadata stored in the catalog. We prototyped this by determining the percentage of the possible metadata fields populated. In addition, we were able to weigh specific fields more in order to improve the significance of the metric. For example, specifying a responsible person has a higher weight than specifying a version number.

A second participant suggested that not only numbers and strings be outputted as results but also visual results in the form of charts. Due to the scope, we could not elaborate on

this. In principle, however, letting the user select from a set of prefabricated diagrams would be conceivable.

The question also arose whether metrics can be applied to data not inventoried in one's catalog. For example, it can be applied to data offered by another company that operates its own data catalog. We suggested that the metadata from the other catalog be transferred to our own. This mechanism is already established in ODPs in the context of data harvesting.

Overall, participants were satisfied with the functionality but noted that writing the metrics in JavaScript was only accessible to experts. They referred to Blockly[1] and similar visual programming languages. These could enable a broader audience to generate their own metrics.

Participants also suggested whether DIVA could be used to identify potential risks related to the use of data. We have developed this idea further and published it [Teb18, Paper II.]. There, we describe the previously discussed **design knowledge in form of a method** for making metrics in data catalogs more flexible. This design knowledge may be used to develop data catalogs and will be discussed later when we demonstrate our DPs. We also describe how a federated architecture approach can be used to exchange metadata with each other. Risk metrics can then be developed and calculated based on internal and external metadata. For example, we describe the case where external data is no longer available, and there is a risk that internal business processes could be negatively impacted.

Conclusion

As part of the second iteration, DIVA 1.1 was developed and extended to include the function of adding one's own metrics based on the existing metadata (R2.1). The implementation is not limited to one metric type but can be used, for example, to generate quality or risk metrics. This flexibility allows DIVA for a more detailed data quality or risk assessment. This idea was published as under the title *Risikobewertung in Datennetzwerken* at DACH Security 2018 [Teb18, Paper II.]. We consider this concept as its own result (R2.2). A summary of the second iteration can be seen in Figure 5.2.

5.3 Iteration 3

This iteration was initiated by the pharmaceutical company, in which also the first iteration was carried out. Much data was generated on the employees' devices and was, in some cases, only stored there. Over time, data accumulated to which only the respective employee of the device had access. For the latter, it was also not necessarily apparent that he or she had data that could interest others. Therefore, this iteration aimed to create a solution that would help find interesting data on employees' devices and make it discoverable by others in the organization if needed. This project was also motivated by the findings of the first iteration. There, we discovered that the DIVA data catalog can only be used meaningfully by automating the inventory and profiling of data. For this reason, we developed a solution that partially automates both aspects. On the one hand, the software had to search employees' computers for data and extract interesting metadata. On the other hand, after

1 `https://developers.google.com/blockly` [Accessed: July 11, 2023]

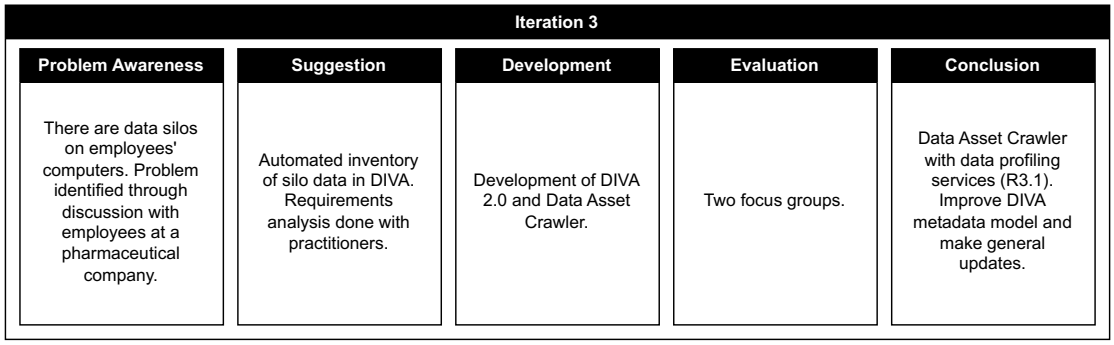


Figure 5.3: Summary of the DIVA development iteration 3

the device's owner had given his or her consent, the metadata had to be transferred to the central DIVA instance. Transferring metadata lets company colleagues find data of interest through the data catalog and request access when needed.

Development

The Data Asset Crawler (DAC) was developed as a companion application for DIVA that can be installed on employees' computers. The application itself was implemented using Electron[1] so that it can be used on a wide variety of operating systems. The DAC maintains its local index of the file system and monitors changes. The software can be configured so that users can decide which folders to index. Likewise, one can decide whether only specific file types should be considered for indexing.

For specific data types, analysis services were implemented to automatically determine further metadata for the data catalog. The focus was on texts, tables, and images. With the help of Apache Tika[2], the raw text was extracted from various text formats. Then, keywords were extracted using the PositionRank algorithm [Flo17], Also, information on the number of words, sentences, and the like was collected. For tables, the schema was extracted, and simple statistical evaluations were performed. This includes, for example, the calculation of average values or frequency distributions. For images, object recognition and generation of captions were implemented. The user can view the indexed files in the application and synchronize the metadata with DIVA as desired. Figures A.10 and A.11 show screenshots of the application.

In addition to developing the companion application, DIVA was further enhanced. On the one hand, the metadata model and the user interface were extended in order to be able to store the new kind of metadata generated by the DAC. Likewise, dependencies were updated, and the available Hypertext Transfer Protocol (HTTP) API was rebuilt. Therefore, we made a major version jump to DIVA 2.0.

1 `https://www.electronjs.org/` [Accessed: July 11, 2023]
2 `https://tika.apache.org/` [Accessed: 11 July 2023]

Evaluation

The application was presented to two focus groups. The first focus group consisted of pharmaceutical company employees working in a data-driven department. The group found the solution very interesting and could imagine that it would be easier to find data, even if only on their laptops. One participant referred to the Google Desktop tool, which Google had discontinued but which he used intensively. Google Desktop allowed a computer-wide search in supported files such as PDFs or emails. There was no reasonable alternative then, and the DAC would have been a suitable replacement. According to another participant, the automated metadata profiling allows for a much better search than the operating system offers.

The second focus group consisted of management-level employees from the department described above. They were very open to the basic idea but were pessimistic about further evaluating the application with real users. The main reasons for this were twofold. The first reason was privacy concerns. It was doubted that the data protection officer would approve an application that scans data on computers and transfers information about it to a central system. Even if the application could be configured, there was concern that employees' private data could be added to DIVA. The second reason was a new company-wide data strategy unveiled just days before the focus group. This strategy states that no work-relevant data can be stored exclusively on employees' laptops but must always be synchronized with existing cloud systems. Therefore, assuming data exclusively on individual employee devices would no longer be necessary. The focus group suggested building the functionality to automatically discover metadata directly into DIVA and create cloud systems connectivity. Further evaluation of the DAC was therefore stopped.

Conclusion

In this iteration, the DAC, a DIVA companion application, was developed (R3.1). DIVA has been updated accordingly to work together with the DAC. The evaluation revealed that the idea underlying the DAC will not be pursued further in the context of the pharmaceutical company. Nevertheless, some aspects of value were elaborated. For instance, the automated data profiling of metadata is an important extension for the DIVA data catalog. A summary of the third DIVA development iteration can be seen in Figure 5.3.

5.4 Iteration 4

Another iteration was started after the previous iteration had to be interrupted in the evaluation phase. The previously obtained feedback about the integration of automated metadata discovery was implemented directly in DIVA. The pharmaceutical company has several departments dedicated solely to collecting and analyzing study data and their analysis programs. An analysis program is a software code that processes the study data to extract findings. However, due to the large amount of study data and analysis programs, it is difficult for staff to identify responsibilities or contexts. This difficulty exists because contextual knowledge about the study data and analysis programs is collected in multiple places. For this reason, DIVA was extended to inventory the study data and analysis programs. Furthermore, related data and programs were linked in DIVA and, if possible, the study data quality should be determined.

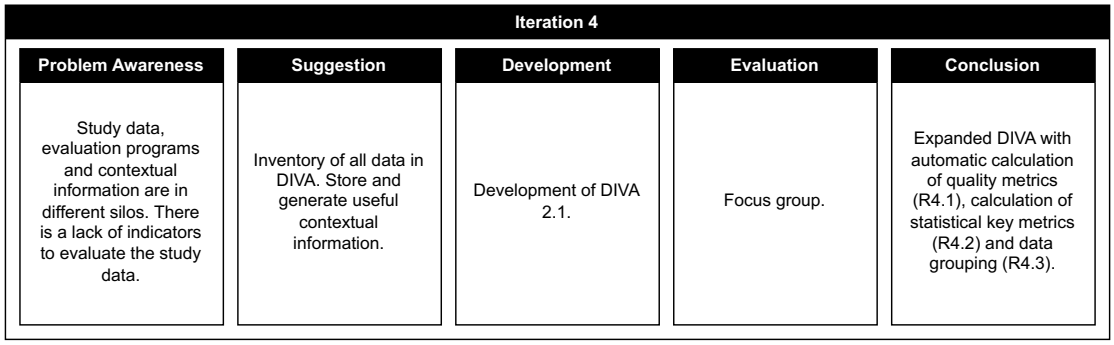


Figure 5.4: Summary of the DIVA development iteration 4

Development

DIVA was first extended to enable the relevant data to be sent to the backend via the web browser. DIVA then automatically creates new entries for the files and starts obtaining the metadata. The study data are files in a particular native format. These are tabular data, so the service developed in the previous iteration for extracting metadata could be reused. The service was extended to include further statistical analysis. Also, simple quality metrics were implemented. For example, they show if the table has empty elements and the data type per column is always the same. An impression of the application at that time can be gained in Figure A.12. Furthermore, a function was added to group inventoried data in DIVA. This feature behaves similarly to a folder in which one can put elements. This way, analysis programs, and study data can be linked in DIVA. The groups created this way are full-fledged entries in the data catalog, can have metadata such as a title, description, and responsibilities, and can be grouped themselves.

Evaluation

In an initial evaluation round, a focus group was formed. In addition to the colleagues with whom we had intensive exchanges during the development phase, there were also people from other company departments. DIVA and its functionalities were presented to the group. The feedback was consistently positive. The focus group sees data catalogs as an important means of improving the company's overview of existing data. Concerning study data and analysis programs, more automated analyses were desired. For example, it would be interesting to know whether the evaluation programs have a high code quality and follow style guides.

Another feedback we got was that simply transmitting the data to a central server and extracting the metadata from there can be challenging. Even when the data remains internal to the organization, it can be extremely sensitive. It was suggested that alternate methods be explored, such as extracting the metadata before sending it.

Conclusion

Functionalities developed in the previous iteration for extracting and generating metadata were integrated directly into DIVA. They were also extended to identify specific relevant metadata in this context. So, we focussed on automatically calculating data quality metrics

(R4.1) and statistical key metrics (R4.2). A new feature in DIVA also allows staff to group inventoried data (R4.3). A summary of the fourth iteration can be seen in Figure 5.4.

5.5 Iteration 5

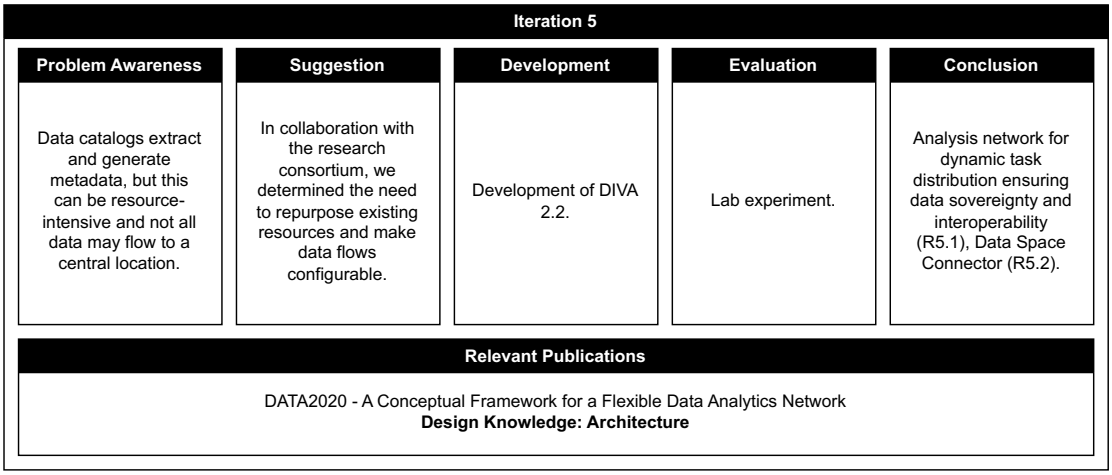


Figure 5.5: Summary of the DIVA development iteration 5

This iteration was launched as part of a research consortium that mainly studies topics related to sovereign data exchange. The pharmaceutical company was also part of the consortium and motivated our iteration. The main focus was on combining the advantages of cloud and edge computing. In particular, it should be explored how tasks can be dynamically distributed across existing computing resources while preserving data sovereignty. Together with the participants, we identified that the challenges addressed here also apply to data catalogs. Data catalogs are a central component that can automatically extract and generate metadata from data. In many cases, however, the data must inevitably flow to the data catalog so that it can perform the analyses. As a result, users lose control over their data. We identified this during our prior iteration evaluation with the pharmaceutical company. Also, the mass of data that the data catalog has to inventory and thus analyze should be considered. Data profiling can bind many resources. For this reason, a solution had to be developed that allows data catalogs to reuse existing resources in the cloud and on the edge. It was important to attach conditions to the tasks, e.g., to limit data flow to trusted computers or to select special computers for particular tasks.

Together with the participants of the project, it was also decided that the data catalog should be able to connect to data spaces that follow the Industrial Data Space (IDS) reference architecture [Bor19]. The IDS aims to create a data space where companies can exchange their data securely, sovereignly, and decentrally. Collaborative rules are to ensure that the data space can function without a central control authority. At the time of this development iteration, the connection to the IDS required a lot of technical know-how and domain knowledge. It was assumed that interested parties would not implement their own connector to join data spaces. Therefore, the idea came up to extend the DIVA data catalog with a Dataspace Connector (DSC) compliant to the IDS reference architecture.

Thus, DIVA could automatically extract and generate metadata in a sovereign analysis network and support sovereign data exchange between companies.

Development

This project added a workflow engine to DIVA. It allows individual steps in extracting metadata to be orchestrated in a network of computers. The basis here is a lightweight version of Kubernetes[1] with the workflow engine Argo[2]. Computers that are to be connected to form an analytics network only need to have an instance of K3s[3]. Through the K3s instance, a basic container image is run that brings all the computers into a shared peer-to-peer network. The peer-to-peer network is based on the InterPlanetary File System (IPFS)[4], which both bootstraps the network and orchestrates the virtual network over the TCP/IP layer. Arbitrary data can be exchanged over the virtual network using unique content identifiers. Our solution uses this mechanism to deliver both data and code. Since data does not have to be stored centrally, sensitive data can also be analyzed on the user's own or trusted computers. Data is exchanged directly on a peer-to-peer basis. The unique content identifier is not only the locator but also the key for accessing the data. Thus, only participants with the key can access the data. Workflows can be adapted to one's own needs by a flexible tagging system. This allows, for instance, to choose only machines with particular characteristics, such as a specific physical location or a particular CPU or GPU. In sum, we developed **design knowledge in the form of an architecture** that can be used when implementing data catalogs. Further details can be found in our publication [Teb20, Paper III.], which is part of this thesis.

Second, the DSC was implemented in DIVA. Users can upload files to DIVA and make them available via the connector. For example, a small set of usage policies can be configured to determine when the data may only be used (see Figure A.15). The implementation is strongly based on the IDS reference architecture model 3.0 [Bor19].

Evaluation

The evaluation was performed in an experimental laboratory setting. The implemented software was installed on several computers, and metadata has been collected successfully. Details can be found in the publication [Teb20, Paper III.]. The solution has been shown to work and to distribute the load of metadata discovery evenly across the network on existing resources. It could also be shown that data is only distributed to previously declared trusted machines. However, it became clear that software maintenance and deployment are non-trivial. Due to limited resources, this solution could not be further developed in future iterations of DIVA. However, the basic feasibility and usefulness of the solution could be demonstrated.

The evaluation of the DSC also occurred in a laboratory setting. A partner from the research consortium provided a second connector, which requested data from the

1 `https://kubernetes.io/` [Accessed: July 12, 2023]
2 `https://argoproj.github.io/argo-workflows/` [Accessed: July 12, 2023]
3 `https://k3s.io/` [Accessed: July 12, 2023]
4 `https://ipfs.tech/` [Accessed: July 12, 2023]

DIVA connector. The goal was to see if these two completely independently implemented connectors could talk to each other. We got a positive result here. The connectors were able to exchange data and usage policies with each other successfully.

Conclusion

In this iteration, DIVA was extended to include a flexible analysis network by distributing the tasks of extracting and generating metadata (R5.1). A particular focus was placed on the reuse of resources and the preservation of data sovereignty. The results were also published in the publication *A Conceptual Framework for a Flexible Data Analytics Network* at DATA2020 [Teb20, Paper III.]. This paper is part of this thesis. We also implemented a DSC that allows data to be shared securely and sovereignly from within DIVA (R5.2). A summary of the fifth iteration can be seen in Figure 5.5.

5.6 Iteration 6

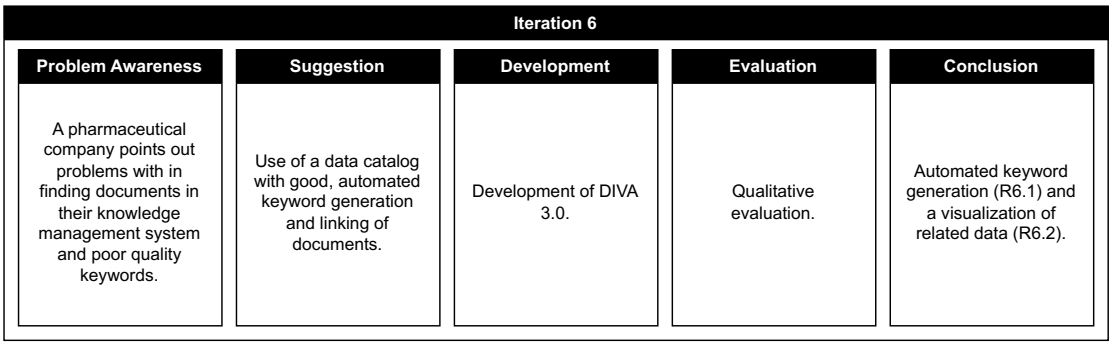


Figure 5.6: Summary of the DIVA development iteration 6

This iteration was again carried out in collaboration with the pharmaceutical company. It was explained to us that employees often have to work with medical documents such as files, prescriptions, or patient information leaflets. All of this happened in a particular corporate knowledge management application. This application had over one million page hits and over 6,000 users globally. Keywords played an essential role in the system when finding relevant and linked documents. Users created them manually, who did not follow any ontology but chose keywords based on gut feeling. The system's search function was unsatisfactory for the users because the keyword search virtually did not work. Even linked documents could not be found because they have different keywords. Several proposed solutions resulted from the problems mentioned above. First, the data catalog DIVA was used to inventory the documents. The generation of the keywords was to be automated. Based on an ontology provided by the company, an attempt was made to standardize keywords. In addition, linked documents were visualized via a graph so that users could jump from document to document.

Development

DIVA was enhanced in many aspects in this iteration. First, based on the results of the last iteration, a new workflow engine was integrated into DIVA. This update was necessary because too few resources were available for further development and maintenance of the

previously proposed solution. The workflow engine Apache Airflow[1] was integrated. The previous services for metadata extraction could be adopted entirely, except for minor adjustments. The project focused, in particular, on the automated extraction of keywords for medical documents. For this purpose, a specialized service was implemented, which extracts keywords from the raw text and tries to match them with the provided ontology. By applying the Levenshtein distance, keywords could also be mapped if they differed slightly from the term in the ontology. Thus, on the one hand, the standardized keywords, and on the other hand, the keywords detected automatically by the algorithm were stored. The grouping function was reused to link documents. Thus, documents that are related to each other can be grouped by the users. Likewise, a visualization was implemented so that users can navigate through the graph (see Figure A.14). Overall, the web application has been renewed, building on Vue.js[2] and the component library Vuetify[3]. We also implemented visualizations for some of the stored metadata to make it easier for users to interpret them. An impression of what DIVA looks like in version 3.0 can be seen in Figure A.13.

Evaluation

The improvement in the automated extraction of keywords from texts was evaluated iteratively in close cooperation with a pharmaceutical company team. It started with the solution already implemented in DIVA based on PositionRank and the first version of the ontology. Keywords were extracted from some documents and presented to experts to evaluate the solution. They rated whether a keyword was good, neutral, or wrong. Due to mixed results with PositionRank, an alternative solution was implemented based on Rapid Automatic Keyword Extraction (RAKE) [Ros10a]. Here, too, the generated keywords were evaluated by experts. The results were rated better than they were with PositionRank. A further iteration was implemented, which further filters and processes the result of the RAKE algorithm. Thus, keywords were limited to two words and exclusively nouns. Also, a new version of the ontology was used. Another evaluation was performed, which again showed slightly better results.

Conclusion

In this iteration, improvements have been made regarding the functionality of DIVA. One highlight is the automated generation of keywords (R6.1) based on an ontology from the medical field. This automatism has dramatically improved the search for documents. The linking of documents is also worth mentioning at this point. Users can now easily find relevant and related documents through visualization, even across multiple documents (R6.2). A summary of the sixth iteration can be seen in Figure 5.6.

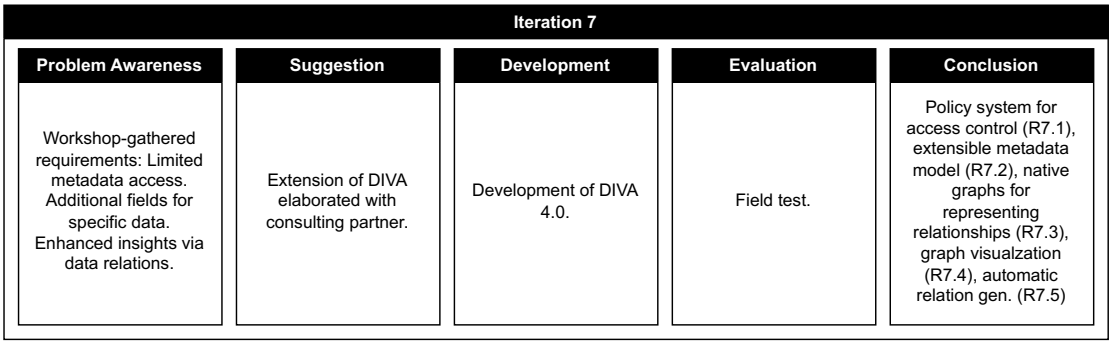


Figure 5.7: Summary of the DIVA development iteration 7

5.7 Iteration 7

This iteration was conducted in cooperation with a consulting partner that offers guidance to cities and municipalities in Germany on topics related to digitization and innovative data use. With this partner, we conducted workshops with cities and municipalities throughout the project. The workshops aimed to introduce DIVA and demonstrate it live. Furthermore, we wanted to see whether DIVA could already address the challenges of the workshop participants or whether DIVA needed to be adjusted. The feedback from the workshops will be briefly described below.

In principle, the cities and municipalities were convinced by DIVA and its idea of making data more accessible to find and use for everyone. In detail, however, there were significant differences in the requirements a data catalog must fulfill. First, there was the issue of access control. All users could find and edit all the data inventoried in DIVA in version 3.x. Several participants in various workshops expressed concerns about this. In practice, it often happens that the mere existence of certain data should not be carried beyond a particular group of people. There were also concerns about metadata being accidentally adjusted by inexperienced users. Even though DIVA tracks every change, it would make sense to allow changes only by responsible parties. However, the cities and municipalities had different ideas about what the access regulations should look like. The requirements ranged from access and editing only by invited persons to almost everyone being allowed to access and edit.

In addition to access to the content, the application's accessibility in general was also discussed with one city. This discussion included the language of the application and its accessibility. DIVA is only available in English, which was a factor against its use by the city. Due to a lack of resources and expertise, accessibility is only rudimentary implemented in DIVA.

Concerning the metadata model, the requirements of cities and municipalities diverged. The workshop participants found the metadata already available in DIVA universally useful.

1 `https://airflow.apache.org/` [Accessed: August 11, 2023]
2 `https://v2.vuejs.org/` [Accessed: August 11, 2023]
3 `https://vuetifyjs.com/` [Accessed: August 11, 2023]

However, there were many requests for new fields to input specific metadata. One city had a lot to do with geodata and wanted to be able to store geo points on a map. Another municipality had an internal project with e-scooters and wanted to inventory the sensor data stored in an existing IoT platform in DIVA.

The last major requirement we identified in the workshops was the ability to represent arbitrary relations between data. DIVA already offered a way to group data and then display it as a graph via a visualization. However, this solution was not based on a native graph in the backend but on a folder structure. Therefore, DIVA lacked some capabilities for representing relations between data. For example, it was impossible to create directed edges between the data or assign a label or weights. A native graph makes it possible to cover many city and municipal use cases. One specific use case mentioned more often was linking data based on similarity metrics to show recommendations in the catalog. Another use case mentioned was the free exploration of available data based on arbitrary relations and via a user interface designed for this purpose.

After completion of all workshops, it was discussed with the consulting partner that DIVA should implement the following requirements. First, fine-granular access control should be implemented in DIVA so that different cities and municipalities can configure their ideas. Likewise, it should be possible to adapt the metadata model to store various additional metadata and thus context. Next, similar data should be linked automatically and be viewable via a graph view. Concerning the localization of DIVA and accessibility, it was agreed not to implement this for the time being, as it was only requested by one city.

Development

First, the graph database Neo4j[1] was added to DIVA. With Neo4j, the many complex relations between the entities can now be mapped natively in DIVA. This feature allows DIVA users to view relationships between data and other entities such as users, services, and more via a graphical visualization of the data network (see Figure A.18). It allows traversing through the graph to identify possible interesting new data. The graph is generated automatically and kept up-to-date. In this way, responsibilities or similar data are linked to each other. DIVA also allows one to create one's own edges with custom labels and data via an HTTP API.

In the process, we also implemented the ability to link similar data with each other automatically. We followed two approaches. The first approach applies to all data types inventoried in DIVA. The similarity is calculated using the keywords. We use a Locality-Sensitive Hashing (LSH) method, which we run as Functions as a Service (FaaS) in Airflow. The similarity score between two LSH hashes is higher when the keywords are more similar. Above a certain threshold, an edge is created between the data nodes in Neo4j, and the score is used as a weight. The second approach uses the exact mechanism but only applies to text files. Again, similarity edges are generated above a certain threshold. When the data is inventoried, the LSH is calculated for both approaches. The comparison with all other LSH for the computation of the scores takes place periodically since this, despite

1 `https://neo4j.com/` [Accessed: August 11, 2023]

employing a BK-tree as the data structure for the LSH, is a time-intensive undertaking.

A policy system was also implemented, which allows fine-granular access control to be set at runtime via a Representational State Transfer (REST) API. The policy system is a separate microservice that can be attached to the other microservices of DIVA using the sidecar strategy. The services in DIVA ask the policy system on every request and then follow its decision. Thus, it is fundamentally similar to policy systems such as the Open Policy Agent[1] or Oso[2]. Unlike the latter, our solution can not only access the request data to make a decision but also execute and combine complex queries in the MongoDB or Neo4j database to decide whether access may be granted. Furthermore, the policies in DIVA are not implemented with a programming language but are modeled declaratively in JavaScript Object Notation (JSON). Each policy is implemented as a JSON document. All accesses in DIVA are initially forbidden, and the policy system checks relevant policies and then informs the calling service whether access can be granted.

DIVA has also been enhanced with the ability to add custom metadata fields. Users cannot customize the basic metadata model to avoid possible incompatibilities with other data catalogs in the future. The creation of new attributes is reserved for the administrators of the respective DIVA instance and can be done via the user interface (see Figure A.17). Whether new metadata fields are only available for certain types of inventoried data can be configured. For example, fields that are only available for data with a specific Multipurpose Internet Mail Extensions (MIME) type can be created.

Additionally, in this iteration, the web application was again updated to handle the new flexibility of displayable data. This update also includes a slight adjustment of the look and feel. Due to many changes in the backend, the web client, as well as the API, DIVA, has been upgraded to version 4.0. For example, geography information can now be displayed on a map requested by cities and municipalities (see Figure A.16). Figure A.19 again gives a good impression of how, for example, the display of statistical analyses has changed in contrast to DIVA 3.0 (see Figure A.13).

Evaluation

For the evaluation, DIVA was deployed by the consulting partner and used in several individual projects with the cities and municipalities. The goal was to get an overview of the existing data by systematically inventorizing it in DIVA. Based on the data inventoried in DIVA, it was to be determined whether the existing data could be used to establish new business areas, offer new services to citizens, or carry out other data-based optimizations in existing processes. We did not accompany the projects, so we can only provide the feedback of the consulting partner here.

In principle, the use of DIVA in the individual projects worked well. Technically, there were only minor problems, most of which we could resolve during the project runtimes. The data was uploaded to DIVA by the employees of the cities and municipalities via the web interface. This process took several days, as there was no automatic process. After uploading the data, DIVA performed a data profiling for known data types. Subsequently,

1 `https://openpolicyagent.org/` [Accessed: July 12, 2023]
2 `https://www.osohq.com/` [Accessed: July 12, 2023]

the consulting partner used DIVA to get an overview of the available data. It was highlighted that DIVA could support this through several capabilities. The automated data profiling was beneficial, as the large amount of data did not burden the city and municipal staff with additional work. The visualizations of the statistical analyses of tabular data were especially complimented here. This feature made it possible to understand better the many available CSV files. The ability to group inventoried data and browse it in a graph was also highlighted as very helpful. The consulting partner used this capability to group data that, when combined, could represent an asset of the city or municipality. In addition to the positive aspects, there were also comments for future developments. For example, much more automation was desired for data profiling. There were many types of data for which we did not provide any analysis, so only a little insight could be generated there. Geospatial data is one example.

Conclusion

In this iteration, features were implemented in DIVA 4.0, adding value for cities and municipalities. A fine-granular policy system for access control was implemented, which can be customized as desired by the operator of a DIVA instance (R7.1). It was also possible to store additional metadata fields in DIVA and to make their existence dependent on other metadata fields (R7.2). The data network was improved by implementing a native graph database (R7.3). During this process, the graphical interface for displaying and interacting with the graph was also improved (R7.4). Finally, we implemented a way of automatically generating relations between similar data (R7.5). A summary of the seventh iteration can be seen in Figure 5.7.

5.8 Iteration 8

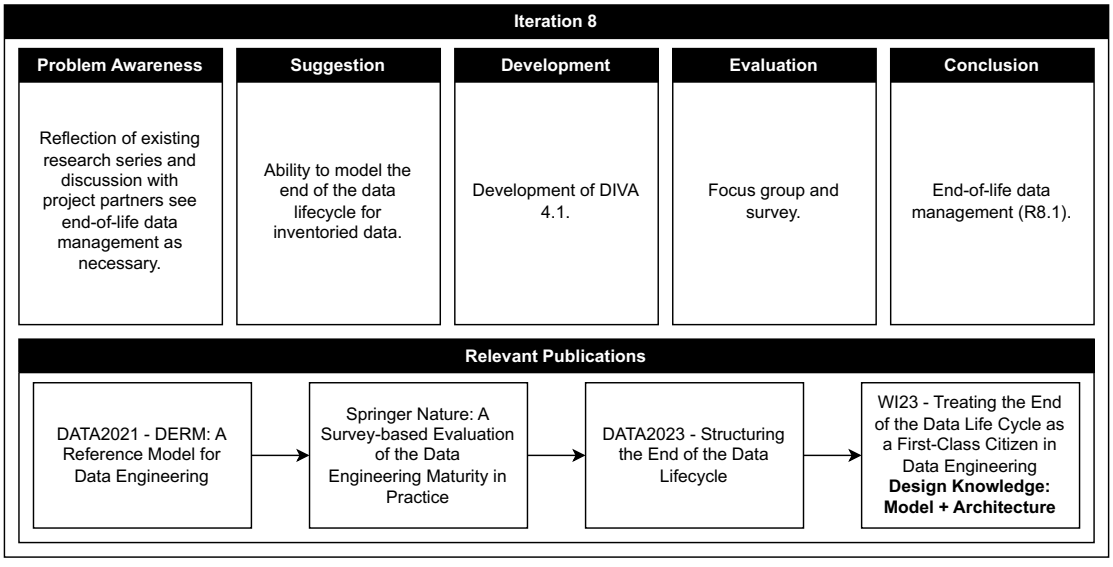


Figure 5.8: Summary of the DIVA development iteration 8

This iteration took place as part of a project with a research consortium. The project's

main goal dealt with creating a data platform in the construction industry with a focus on Artificial Intelligence (AI) applications. One of our tasks in the research consortium was to explore the topic of data sovereignty further. Much work, especially in data spaces, deals with data sovereignty. Often, usage control is addressed as part of this. It provides data owners with solutions to define how their data can be used. As the name suggests, the focus is on the usage aspect of the data. In this project, we decided to focus on the data sovereignty aspect of data deletion. That is, to give users the modeling ability to describe when their data needs to be deleted. This work is supported by a research series conducted in parallel to the project, which identified that the end of the data life cycle generally receives little attention. In the following, we will briefly outline the research series. The papers are to be considered as part of this thesis.

The first publication *DERM: A Reference Model for Data Engineering* aimed to collect and structure an overview of the different phases and perspectives of data engineering [Teb21, Paper IV.]. Here, a reference model for data engineering was developed and evaluated. During the evaluation, it became clear that data engineering communities pay little attention to phases like data planning, creation, and destruction. Since this is a literature-based work, a practical perspective was obtained in a follow-up work. In *A Survey-based Evaluation of the Data Engineering Maturity in Practice*, we surveyed 28 data and software engineering people about the phases and perspectives identified in the previous work [Teb23a, Paper V.]. Of relevance here is the fact that even in practice, the topic of data deletion receives little attention. Thus, little to no standards, processes, languages, and tools related to this topic find their application in practice. Likewise, there needs to be more metadata maintained regarding the end of the data life cycle. Based on the two publications mentioned above, we decided to examine the end of the data life cycle as a research topic. For this reason, we looked at the current state of the art on the topic in the literature in our publication *Structuring the End of the Data Life Cycle* [Teb23c, Paper VI.]. This research identified how the literature currently reflects the topic, the fundamental aspects of data deletion, and how to structure the topic area. The topic has received very little attention. Based on the publications on the topic published in the last 23 years, it was possible to develop a taxonomy that structures the subject area for the first time.

Building on the previously described, a model should be developed to describe the end of the data life cycle. The taxonomy of data deletion served as the basis for developing this model. It should enable data owners to specify precisely under which circumstances their data must be deleted. The idea was also to store this information in DIVA so that all relevant information about data is consolidated in one place. To ensure that a user does not always have to check whether data needs to be deleted manually, automatic mechanisms and their integration into software systems had to be developed.

Development

We extended DIVA with the ability to model the end of the data life cycle for inventoried data. So-called Destroy Claims can be created and managed via the user interface [Teb23d, Paper VII.]. The model is based on the taxonomy of data deletion established in [Teb23c, Paper VI.]. Figures A.22, A.23 and A.24 give an impression of how the implementation looks like in DIVA 4.1. The Destroy Claims can then be used by another application,

called Destroy Claim Agent (DCA), to determine whether data should be deleted. A DCA can perform the deletion automatically or inform users that obsolete data exists. We implemented a JavaScript library to simplify significantly the creation of a DCA. The library is available online on GitHub.[1] More details can be found in our publication [Teb23d, Paper VII.].

Evaluation

A focus group and a questionnaire evaluated the developed model to describe the end of the data life cycle and the implementation in DIVA. Details of the evaluation can be found in the publication [Teb23d, Paper VII.]. A demonstration of the functionality was also performed. Details of the demonstration can be found in the Zenodo repository for the paper.[2] Overall, the results are promising, and the evaluation participants see the solution as an important building block, especially for professional environments such as companies. Also, end-of-life data management in a data catalog is viewed positively.

Conclusion

In this iteration, the Destroy Claim model was developed to support end-of-life data management (R8.1). The specification of the Destroy Claim model is freely available on GitHub.[3] It was implemented in DIVA so users can model whether inventoried data must be deleted. A library was also developed to generate DCAs that can interpret the Destroy Claims and perform the deletion automatically if necessary. The library is available freely on GitHub. We furthermore propose how to implement Destroy Claims and DCAs in a software architecture. The paper *Treating the End of the Data Life Cycle as a First-Class Citizen in Data Engineering*, published at the International Conference on Wirtschaftsinformatik 2023, transfers the results to the knowledge base [Teb23d, Paper VII.]. Here we present **design knowledge in the form of a model and an architecture** that can be used to implement data catalogs. A summary of the eighth development iteration can be seen in Figure 5.8.

5.9 Summary

During the iterations, challenges from practice were brought to us. Solutions were developed in close cooperation with industry partners as part of the data catalog DIVA. Likewise, these solutions were evaluated with the help of practitioners. Different evaluation methods were used, chosen within the limits of the available resources. Our results can be summarized as follows:

- Several versions of the software artifact DIVA were developed throughout the iterations. The software was developed over six years and provides implicit design knowledge. DIVA is available as FOSS on GitHub. We have also developed the DAC software, which allows data to be indexed on local computers and metadata

1 `https://github.com/DaTebe/destroyclaims` [Accessed: August 12, 2023]
2 `https://doi.org/10.5281/zenodo.8046369`
3 `https://github.com/DaTebe/destroyclaims/blob/9a8a6e5e432312175cc439f333c45620924400fc/docs/destroy-claim.md` [Accessed August 14, 2023]

to be extracted automatically. This can then be sent to a central instance of DIVA so others can discover the data. We have also written software that allows Destroy Claims to be interpreted and, if necessary, the claim to be executed. This software is also available as FOSS on GitHub.

- In addition to the software artifacts, various ideas have been encapsulated and published. The resulting peer-reviewed publications add design knowledge to the knowledge base in the form of **models** [Spi18, Paper I.], [Teb23d, Paper VII.], **methods** [Teb18, Paper II.], and **architectures** [Teb20, Paper III.], [Teb23d, Paper VII.] that can help in implementing data catalogs. Publication [Teb23d, Paper VII.] was prepared by publications [Teb21, Paper IV.], [Teb23a, Paper V.], [Teb23c, Paper VI.]. Therefore, these are also seen as part of this thesis.
- We described and collected the DIVA development results to develop our design theory for data catalogs. The specific results are the following: Data Assessment Capability (R1.1), Data Model for Data Goods (R1.2), Management Overview (R1.3), Review System (R1.4), Customizable Metrics (R2.1), Quality and Risk Assessment (R2.2), Data Asset Crawler (R3.1), Calculation of Quality Metrics (R4.1), Calculation of Statistical Metadata (R4.2), Data Grouping Capability (R4.3), Analytics Network (R5.1), Data Space Connector(5.2), Automated Tagging (R6.1), Data Grouping Visualization (R6.2), Policy System (R7.1), Extensible Metadata Model (R7.2), Native Graph DB (R7.3), Graph DB Visualization (R7.4), Automatically Relate Similar Data (R7.5), End-of-Life Data Management (R8.1).

CHAPTER 6

Designing Data Catalogs

We will now discuss our results about the design theory. First, we will investigate RQ1 by presenting our DPs for data catalogs, which are fundamental to our design theory. The following seven sections are each devoted to one DP. First, we discuss the DP in general, clarify terms, and position it in the context of data catalogs. The latter is done with the help of the identified literature that supports the DP. For each DP, we will also answer RQ2 accordingly by discussing the DFs that support the implementation of the DP. DFs are first discussed in general terms and then provided with implementation recommendations and application examples. Finally, we will demonstrate the overall design theory for data catalogs and all its components and thus address the MRQ.

6.1 DP1: Principle of Automation

Table 6.1: DP1: Principle of Automation

Design principle title	**DP1: Principle of Automation**
Aim, implementer, and user	To efficiently and cost-effectively provide an up-to-date and comprehensive data catalog inventory with accurate, error-free, and high-quality content (aim) for its users (user).
Mechanism	Automate as many processes as possible that are necessary for the seamless operation of the data catalog.
Rationale	Due to the extensive amount of existing data, it is impossible to manually maintain the data catalog content efficiently.
Extracted from	R2.1., R3.2, R4.1, R4.2, R5.1, R6.1, R7.5
Supported by	[Ade17; Aik20; Car11; Fiz20; Frt21; Hal16; Hod21; Kim21; Kir19a; Lab20a; Mar13; Neu16; Noy19; Sch17; Sen18; Tyg16; Urb22]
Design Features	• DF1.1: Automated Inventory • DF1.2: Automated Metadata Gathering

This section discusses the first DP for data catalogs. Table 6.1 shows an overview of the DP. Automation plays an essential role in many areas of life. One of the most important

benefits is the increase in efficiency. Automation allows processes to run faster and reduces errors by eliminating the need for human actors. Removing the human actor can also increase accuracy, as the automation is more precisely tuned to the task and does not lose attention over time. This can lead to a higher quality result. All in all, these factors impact possible cost savings through saved labor costs, that fewer errors must be corrected, or that the result is of higher quality. Automation does not have to mean that jobs for employees have to be cut, but that capacities can be allocated for higher quality work [She17].

Automation also plays a vital role in the context of data catalogs. Labadie et al. see automation as "[...] one way of optimizing data catalog implementations, as well as their maintenance." [Lab20b, p. 209]. Within a data catalog, numerous processes, such as the collection and inventorying of metadata, present significant challenges when executed manually. In fact, manual execution of these tasks is often deemed impractical. The literature deals here in particular with the automated filling of the data catalog by harvesting, that is, crawling of other data catalogs or the internet [Car11; Hod21; Kim21; Kir19a; Mar13; Noy19; Sch17]. Another significant area where automation can make a difference is in the linkage of data, enhancing the search functionality. This is a major effort that is best handled through automation. Once the data catalog is populated, duties such as maintenance, aggregation, and generation of new metadata are required, which can be supported by automation [Ade17; Fiz20; Frt21; Hal16; Tyg16]. Likewise, the metadata quality in the data catalog plays a crucial role. Only with high-quality metadata can the data catalog fulfill its task well (see Section 6.7.1). For example, support for this can be found through automated metadata quality checking [Aik20; Kir19a; Neu16]. Automation can also be important in localizing metadata to address a broader audience [Urb22].

6.1.1 DF1.1: Automated Inventory

To implement DP1, provide mechanisms to inventory relevant data and other entities automatically.

Automating the inventory of relevant data in a data catalog is one of the most important features that should be available. The high frequency with which new data is generated or updated makes manual maintenance time- and resource-intensive. Maintaining the data catalog can also become impossible depending on the amount of new data constantly generated. Manual inventory should only be applied in cases where an automated solution is not applicable. For example, this can be when a data offering does not yet exist but should already be announced in the data catalog. Additionally, data and other entities, such as publishers or services, should be inventoried automatically.

Implementation Recommendations

Automated data inventory can be broadly divided into push and pull strategies.

Pull strategy: In this strategy, the data catalog independently queries other data catalogs or technical systems and actively searches for data to be inventoried. The data catalog must be able to operate the appropriate interfaces and interpret the retrieved data. The results are then transferred to its database. This is also called data harvesting in the area of ODPs. Implementing this strategy is ideal if only a few systems need to be connected and the data landscape does not change regularly. Otherwise, the situation

arises that the data catalog must be constantly expanded. This strategy is also suitable if existing systems cannot be adapted to forward new data offers to the data catalog.

Push strategy: With this strategy, the responsibility for the inventory of the data catalog is handed over to external systems. These must be able to understand the API and the data model used by the data catalog (see also Principle of Interoperability). It must be ensured that these systems can be trusted and do not flood the data catalog with incorrect data. This strategy must be used if the data catalog cannot identify and inventory the data in a system, for example, for technical reasons. This is the case, for instance, if the data is located on an employee's device. Here, standalone applications are required to scan the system and then pass on the data to be inventoried to the data catalog. This strategy should also be used when new persistence technologies are regularly addressed. Specialized applications can be developed for this purpose.

In addition to a purely technical decision, the selection of strategies can also be made from an organizational perspective. The choice of strategy implicitly influences the distribution of responsibilities. While the pull strategy tends to shift responsibility toward the data catalog, the push strategy shifts it to the external systems. It should be mentioned that a mixture of strategies can occur or even be unavoidable in complex systems.

Application Examples

The EU ODP is a concrete example that implements the pull strategy. The portal regularly communicates with over 80 other ODPs and updates its own data inventory [Kir19c]. DataHub and OpenMetadata have implemented push and pull strategies.[1, 2] As part of the DIVA Iteration 7, a pull strategy was developed for DIVA to inventory data from IoT Platforms. An automated inventory using the push strategy was implemented in the DIVA Iteration 3. There, the DAC was developed, which can be used to automatically send metadata from personal devices such as laptops or desktops to DIVA (see Figure A.11 and A.10) (R3.1).

6.1.2 DF1.2: Automated Metadata Gathering

> *To implement DP1, provide mechanisms that automatically discover relevant metadata for inventoried data.*

"The ability to automatically ingest metadata from existing data systems into a DC [Data Catalog] is another important issue [...]" [Jah23, p. 93]. A data catalog can only perform its primary function well if metadata is available in sufficient quantity and good quality. Due to the large amount of data that such a data catalog can inventory, it makes sense to automate the process of populating it with metadata. "Metadata collection is expensive unless it is automated or at least partially automated [...]" [Jef20, p. 128]. Generally speaking, there must be an instance that extracts the metadata or generates a new one from the data by doing data profiling. Data profiling results can be inherently diverse, as they depend on the data. They range from statistical analysis to generating

1 `https://datahubproject.io/docs/metadata-ingestion` [Accessed: May 13, 2024]
2 `https://docs.open-metadata.org/v1.3.x/deployment/ingestion` [Accessed: May 13, 2024]

textual summaries and data classification. This amount of metadata to be profiled would be laborious to maintain manually and offers great potential for human error. Automation is therefore recommended and necessary for providing a well-functioning data catalog.

Implementation Recommendations

Data profiling in data catalogs can be complex and require many individual processing steps that build on each other. For example, for text, it may be necessary first to extract the raw text from a native format, then determine the language, and finally extract keywords. They should be implemented as individual steps to allow for different forks in the execution. With the raw text, one can still determine statistical values afterward. After detecting the language, sentiment analysis can also be performed with the raw text. It makes sense to use a workflow engine that orchestrates these functions. In addition to the orchestration, some engines also allow one to adapt the workflows to one's own needs via an editor [Afg22; Ber09; Rez19] (see also DF 2.4)

There are several aspects to consider when selecting or implementing a suitable workflow engine. In general, workflow engines can be classified according to the centralized or decentralized nature of the data, the control, and the services [Sto15]. Figure 6.1 visualizes the dimensions for better visibility. Stojnić sees an ideal workflow engine as distributed in

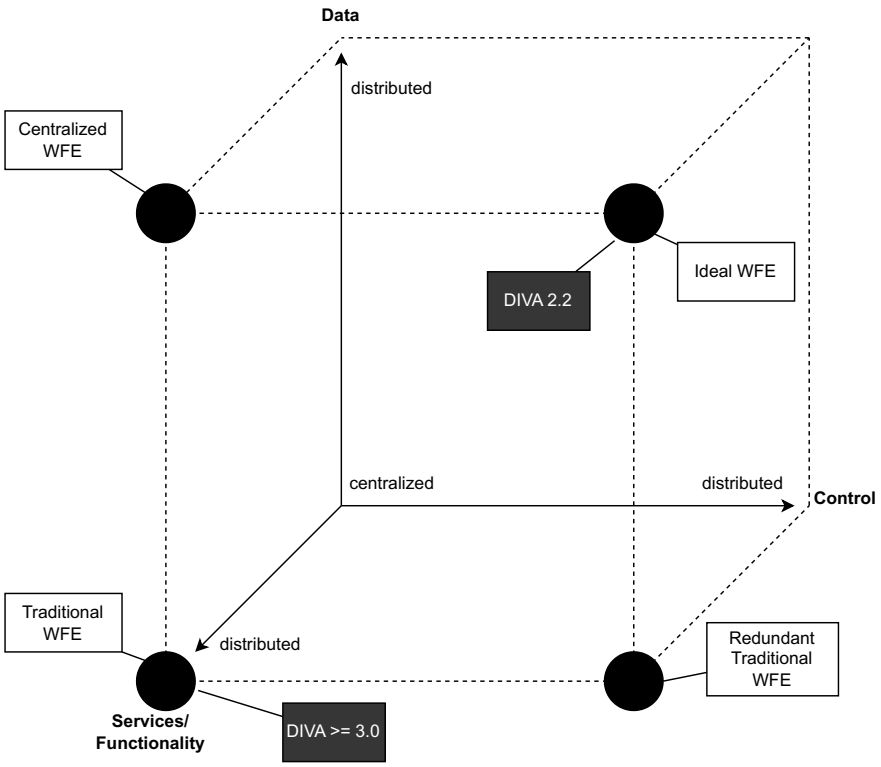


Figure 6.1: Positioning of the DIVA workflow engine implementations in the classification dimensions according to Stojnić [Sto15]

all three dimensions. It should be noted, however, that this imposes complexity and the

usual challenges of a distributed reliable system. If this is unnecessary, e.g., for scalability or fault tolerance reasons, more straightforward solutions are recommended.

In addition to choosing the appropriate workflow engine, it also plays a role in determining who is ultimately responsible for the data profiling. The most obvious possibility is that the data catalog provides a workflow engine with analysis functions and triggers suitable workflows when new data is inventoried. Another option would be for external applications to have their own workflow engines and submit the results to the data catalog. This solution is particularly applicable if the data catalog and its workflow engine can not access the corresponding data to extract the metadata. Also, a federated solution can be provided so that the actors can determine which metadata they want to communicate to the data catalog. This can be particularly relevant to maintain sovereignty over one's data. For example, consider the case where data resides on employees' laptops. An application can access this data locally, extract metadata, and send the results to the data catalog. The data catalog cannot usually access the laptop's file system. In addition, there could be data on the devices where parts of the metadata are not allowed to flow into a central data catalog. Examples of this might be contract or salary data.

Application Examples

OpenMetadata has a few simple metadata extraction functions built in, which can be used, in particular, on tabular data. Profiling takes place in simple workflows.[1] In DataHub, there are ready-made classification algorithms to simplify the filling of metadata fields.[2]

In DIVA Iteration 5, a solution with decentralized data management, decentralized service execution, and distributed control was developed and implemented as a prototype (R5.1). The architecture and technology details are in publication [Teb20, Paper III.]. According to Stojnić, this is an ideal workflow engine. However, due to the complexity of the deployment, a more straightforward solution was chosen for DIVA 3.0 using Apache Airflow. We only distributed the service and functionality dimensions there, while the data remained centralized. According to Stojnić, this is a traditional workflow engine [Sto15]. Figure 6.1 shows where DIVA 3.0 is positioned. Even though this solution is not the ideal one according to Stojnić, it was much easier to handle considering the small size of the DIVA development team of mainly two people. As an example of outsourcing data profiling to external applications, consider the DAC from DIVA Iteration 3. While it did not have its own workflow engine integrated, it could extract metadata for a limited set of file types and send it to DIVA.

Regarding content, several data profiling tasks were automated in DIVA. In DIVA Iteration 2, we have developed a solution that allows new metadata, such as risk assessments, to be calculated automatically using the existing metadata (R2.1). More details can be found in our publication [Teb18, Paper II.]. Next, in DIVA Iteration 3, simple analyses were performed on text, table, and image data to obtain metadata (R3.2). In DIVA Iteration 4,

1 `https://docs.open-metadata.org/v1.3.x/connectors/ingestion/workflows/profiler` [Accessed: May 13, 2024]

2 `https://datahubproject.io/docs/metadata-ingestion/docs/dev_guides/classification/` [Accessed: May 13, 2024]

the analyses were extended to obtain more complex statistical analysis and quality metrics (R4.1, R4.2). DIVA Iteration 6 took DIVA major steps forward in automated keyword detection, particularly in the context of medical data (R6.1). Finally, DIVA Iteration 7 focused on automating the linking of similar data to better capture context (R7.5).

6.2 DP2: Principle of Flexibility

Table 6.2: DP2: Principle of Flexibility

Design principle title	**DP2: Principle of Flexibility**
Aim, implementer, and user	To provide a data catalog that can be adapted to different contexts, environments, and future challenges (aim) by the users (user).
Mechanism	Allow customization of components and features.
Rationale	Due to the highly individual requirements demanded of a data catalog, there is no one-fits-all solution. Having the data catalog code itself customized creates unnecessary costs, creates artificial hurdles, and delays the adoption of a data catalog.
Extracted from	R2.1, R5.1, R7.1, R7.2, R7.3
Supported by	[Bas18b; Cza17; Lab20a; Oli16b; Sha05; Tzi21]
Design Features	• DF2.1: Extensible Metadata Model • DF2.2: Creation and Customization of Metrics • DF2.3: Customizable Policy System • DF2.4: Configurable Workflow Engine

This section discusses the second DP for data catalogs. Table 6.2 shows an overview of the DP. Today, changes in the environment to which companies and other entities must respond are of high frequency. The market is constantly in flux, bringing a rapid succession of new competitors, technological advances, changing customer demands, and new regulatory rules. These changes are sometimes so disruptive that entire strategies and process flows must be adapted to the new situations [Maj18; Mey93]. Companies are, therefore, dependent on responding to these challenges. One possibility to achieve this is through flexibilization, which is generally called: "Ability to deal with uncertainty" [Rei09, p. 115]. Only through flexibilization of one's structures and processes to adapt to new requirements can one maintain a competitive advantage or even create one in the first place [Fie91]. However, this flexibility is necessary at the organizational level and extends to the technological level. Only if the underlying technical systems are flexible corporate structures based on them can also be flexible [Nel97].

Knoll and Jarvenpaa, therefore, rightly see flexibility as "[...] a critical design criterion for ensuring alignment of IT to organizational requirements in dynamic environments." [Kno94, p. 11]. On a technological level, flexibility can be defined as "[...] ability to adapt to both

incremental and revolutionary change in the business or business process with minimal penalty to current time, effort, cost, or performance." [Nel97, pp. 77-78]. Flexibility is further subdivided into the dimensions of functionality, use, and modification [Kno94]. These three dimensions describe the flexibility of information systems in terms of transferability to new environments, usability in the context of new use cases, and rapid adaptability.

Providing flexibility also generates costs that must be profitable [Geb06; Sch11]. Here, assessing which degree of flexibility makes sense for one's project is necessary. A complicating factor here is that a software system's flexibility is challenging to measure [Ede06], and whether one has achieved its goal can be unclear.

Therefore, it is not surprising that data catalogs, which can take on a central role in companies and are also exposed to the environment and the associated uncertainty, benefit from flexibilization. This is mainly because data catalogs' requirements and usage scenarios are very broadly positioned. On the one hand, there is an almost unmanageable amount of data formats, persistence technologies, and communication protocols. Depending on the application area, the data catalog must also be able to integrate more exotic or future technologies. The data catalog can support the extensibility, e.g., by a plugin system [Bas18b] or open annotation possibilities [Sha05]. Making the data catalog available as open source also allows adaptation to future challenges and can be a reason for deciding on a specific data catalog software [Tzi21]. Also, a data catalog is not deployed in an empty workspace but must position itself in an existing data ecosystem. Therefore, it must be able to adapt to its environment. So, more than technological flexibility is required; flexibility in terms of content is also required. For example, the data catalog must be able to map existing data governance structures or adapt user and profiling workflows to existing processes. Similarly, fine-grained control for access in collaborative scenarios is an aspect here [Cza17]. Flexibilization of data catalogs in general is also supported by the studies of Labadie et al. in which "[...] nearly all participants stated that they envisioned to implement a single-catalog environment, powered by a dedicated data catalog solution, that they would either configure or customize." [Lab20a, p. 206].

6.2.1 DF2.1: Extensible Metadata Model

To implement DP2, provide mechanisms for users to store new metadata attributes in the data catalog.

Data catalogs are used in various contexts. Situations arise in which a data catalog must be able to store highly diverse metadata. Therefore, a data catalog should allow adaptations to the metadata model. This capability allows users to adjust the catalog independently to their needs without requiring a new implementation.

Implementation Recommendations

Only privileged users should be able to adjust the data model to avoid uncontrolled growth. Adding new attributes is possible at any time without much effort. Care must be taken when removing or updating attributes. First, determine where the attribute is already being used. For example in metrics (see DF 2.2) or in workflows (see DF 1.2). Now, one has to decide whether to turn off the corresponding metrics and workflows until they have been revised or prohibit such attributes' deletion. Updating attributes has the same effect

as deleting an attribute and creating it again. The effect on the remainder of the system must be considered here. Deleting and updating should be restricted for attributes that are part of a standardized metadata model. This is a consequence of the Principle of Interoperability. If attributes from the standard are changed, other data catalogs and systems may no longer be able to fully interoperate with the data model. The data catalog must protect standard attributes from modification by its users or be able to map possible customizations back to the standard. The adaptation of the data model can be extensive. One must determine how the new attribute is named, what type of data it holds, and what type of inventoried data should be displayed. In order to support the users, the adjustment to the data model should be made via a user interface. It can also be used to visualize possible consequences resulting from the changes.

The technical implementation can be very different. In literature, the flexible storage of metadata in data catalogs in the form of annotations is mentioned [Sha05]. In DIVA, each attribute is defined as a single JSON schema. Each schema contains the name, type of attribute, and other information. It also models where the schema is to be attached to the overall model, whether the attribute is visible or editable, and which visualization is to be applied to the client. DIVA assembles an entire model from the individual schemas. Since a schemaless database is used to persist the metadata, nothing else must be considered. Only when deleting or updating attributes is it necessary to intervene. When it comes to mapping complex relations between data and other entities, the use of a native graph database can be helpful. Here, new types of relations can be added much more quickly than would be the case with, for example, a relational database.

Application Examples

The EDC Federated Catalog, DataHub, CKAN, and DIVA all offer the option of extending the metadata model. DataHub offers the possibility to extend existing parts of the model or to implement new aspects.[1] In CKAN, the model can be extended using a plugin.[2] The EDC Federated Catalog implements DCAT but can be extended flexibly.

As part of DIVA Iteration 7, the requirement was formulated that the administrators of the data catalog should be able to add further metadata fields via a user interface. It should also be configurable for which entities and data types the new metadata field is made available. We implemented this solution in DIVA 4.0 (R7.2). Part of the user interface for the administrator can be seen in Figure A.17. Also, a native graph database has been added to DIVA in this iteration, allowing new types of relations to be added easily (R7.3).

6.2.2 DF2.2: Creation and Customization of Metrics

> *To implement DP2, provide mechanisms for users to generate their own metrics based on the metadata stored in the data catalog.*

In addition to adding custom metadata, the ability to extend the metrics contained in

1 `https://datahubproject.io/docs/metadata-modeling/extending-the-metadata-model` [Accessed: May 13, 2024]

2 `https://docs.ckan.org/en/2.9/extensions/adding-custom-fields.html`

the data catalog also plays an important role. Metrics represent a quantifiable measure that users of the data catalog can use to gain insight into the inventoried data. Examples can be Key Performance Indicators (KPIs) or quality metrics (see Principle of Assessment). Users should be able to add their metrics that can be derived from the previous design feature. The metrics existing in the data catalog would not consider the new metadata or only in a highly generic way. For this reason and the previously mentioned general reasons for flexibility, users should be able to add their own metrics.

Implementation Recommendations

Thinking in advance about what data users can access when developing the metric is necessary. The more is allowed here, the more complex the system becomes. The simplest solution is to allow the metric calculation to access only the metadata of the currently displayed data set. This allows the calculation to be performed directly in the client, for example, without sending further requests to the server. A conceivable extension would be to allow access to metadata of other inventoried data. For this purpose, a query language would have to be offered, e.g., to query similar data sets or data sets used in the same project. Access to the results of other metrics would also be conceivable to build new metrics based on them. It should be noted here that the metrics must not be mutually dependent on each other in order to avoid a deadlock. During implementation, it should generally be ensured that metrics do not compute endlessly, whether unintentionally or with malicious intent. After a specific time, the metric calculation should be terminated. On the one hand, continuing to run would unnecessarily tie up resources on the server or client. On the other hand, this can make the data catalog unresponsive. It should also be noted that only some users can query all metadata. Metrics should not be able to circumvent this restriction but should inform the user that the metric cannot be calculated due to a lack of permissions.

If one wants to encourage as many users of a data catalog as possible to develop their own metrics, the implementation should also be possible for non-programmers. One can orient, for example, at the findings from the Visual Programming [Kuh21] and Low Code [Hir22; San19] research.

Users should be given the option to configure the visibility of metrics. This ranges from private metrics to metrics for individual user groups to public metrics for all users of the data catalog. It should also be possible to classify metrics so that they are only displayed on records where they are meaningful.

It can also be helpful to visualize metrics. Comprehensive solutions can provide the user with several possible types of visualizations. Here, for example, one can use the data analysis tool Kibana[1] or Grafana[2] (see Principle of Visualization). Finally, metrics should also be inventoried in the data catalog and be enriched with metadata such as a title, description, and responsible person.

1 `https://www.elastic.co/de/kibana/` [Accessed: June 22, 2023]
2 `https://grafana.com/` [Accessed: June 22, 2023]

Application Examples

In DataHub one can create metrics that indicate incidents.[1] So-called assertions can also be added, which check for certain conditions of the data and metadata.[2] OpenMetadata follows a no-code approach in which test cases for the data can be implemented via the user interface.[3]

As part of DIVA Iteration 2, DIVA has been extended to allow user-defined metrics to be implemented via the user interface (R2.1). The user can access the metadata stored in the data catalog and implement his own computations. Here, the user must be proficient in JavaScript to compute new metrics. The implementation was used in [Teb18, Paper II.] in the context of risk assessments. This feature allows new risk assessments to be added to the data catalog. Concerning risk assessments, see also DF 7.3.

6.2.3 DF2.3: Customizable Policy System

> *To implement DP2, provide mechanisms for users of the data catalog to fine-tune access restrictions to their needs.*

Data catalogs are, as mentioned before, used in different contexts. The rules under which metadata may be created, read, updated, and deleted in the catalog must be adapted depending on the context. These rules can become more complex and evolve.

Implementation Recommendations

The implementation of such a policy system can vary greatly. Therefore, only general ideas can be given here. The solution strategies can differ depending on the database technologies and data formats used. If, for example, JSON is predominantly used to persist the metadata in the data catalog, then it makes sense to use a policy notation specifically for this purpose [Bis16]. The policies' notation should also be chosen so that they can be easily stored in the data catalog's primary database. If, for example, MongoDB or Elasticsearch is used, notations such as JSON-Based Access Control Policy Language (JACPoL) are suitable [Jia17]. Also, on the level of APIs, it should be possible to define policies that regulate accesses. Possible real-world solutions that can be considered here include OPA[4] or Oso[5]. Further inspiration for policy systems can be drawn from major cloud providers such as Microsoft Azure[6] or Google Cloud[7].

1 `https://datahubproject.io/docs/incidents/incidents` [Accessed: May 13, 2024]
2 `https://datahubproject.io/docs/managed-datahub/observe/assertions` [Accessed: May 13, 2024]
3 `https://docs.open-metadata.org/v1.3.x/how-to-guides/data-quality-profiler/test` [Accessed: May 13, 2024]
4 `https://www.openpolicyagent.org/` [Accessed: June 22, 2023]
5 `https://www.osohq.com/` [Accessed: June 22, 2023]
6 `https://learn.microsoft.com/en-us/azure/governance/policy/` [Accessed: June 22, 2023]
7 `https://cloud.google.com/iam/docs/reference/rest/v2/policies` [Accessed: June 22, 2023]

Application Examples

DataHub offers the option of adding one's own access policies.[1] A distinction is made between platform and metadata policies. This way, access can be restricted based on roles and specific metadata values. OpenMetadata has a policy system involving the user and the metadata or operation. A separate policy evaluator analyzes the access and decides whether it is permitted or not.[2]

As part of DIVA Iteration 7, DIVA was extended to include a flexible and fine-granular policy system (R7.1). With this, stakeholders can customize their DIVA instance to their needs and compliance rules. The implementation in DIVA runs on the principle of *deny by default*. Permissions are added additively through individual policies. These are described in JSON and combine the ability to both regulate API access and, at the data level, only allow access to individual fields in the JSON documents. For further flexibility, the policies can access the data of the incoming HTTP request and have read access to MongoDB and Neo4j in DIVA. Thus, the policy's applicability can also depend on the metadata stored in the data catalog.

6.2.4 DF2.4: Configurable Workflow Engine

To implement DP2, provide mechanisms for users to customize workflows to their needs.

Workflow engines can be used for various purposes in data catalogs. One application is the automation of the data profiling (see DF 1.2). Furthermore, workflow engines can perform recurring tasks, such as cleaning metadata. Users of a data catalog may have different requirements for the tasks to be performed. Over time, new tasks may be added, or adjustments to existing workflows may become necessary. For this reason, the workflows in the workflow engine should be customizable.

Implementation Recommendations

It is recommended that ready-made solutions be used for workflow engines. These usually also offer the option of adding, pausing, or adapting workflows at runtime. Examples include engines such as Apache Airflow [Hai22], Kubeflow [3] with the Elyra[4] extension, Galaxus [Afg22], Data Civilizer [Rez19], or KNIME [Ber09].

Application Examples

OpenMetadata provides a workflow engine that the user can configure.[5] The configuration can be done via the user interface.

---

1 `https://datahubproject.io/docs/authorization/access-policies-guide` [Accessed: May 13, 2024]

2 `https://docs.open-metadata.org/v1.3.x/how-to-guides/admin-guide/roles-policies#authorization-framework` [Accessed: May 13, 2024]

3 `https://www.kubeflow.org/` [Accessed: June 22, 2023]

4 `https://www.kubeflow.org/docs/external-add-ons/elyra/` [Accessed: June 22, 2023]

5 `https://docs.open-metadata.org/v1.3.x/connectors/ingestion/workflows/profiler` [Accessed: May 13, 2024]

Since DIVA Iteration 6, DIVA has adopted Apache Airflow as its workflow engine. There, DIVA administrators can customize workflows to suit their needs. Some functions can be done through the easily accessible user interface, like pausing workflows.

Data catalogs often run workflows to extract or generate metadata. This functionality can lead to situations in which data sovereignty or performance aspects must be considered. For example, analyses of sensitive data should not be performed on external computers in the cloud. Complex calculations, on the other hand, should be performed on more powerful machines and not on an employee's laptop. It may also be that the data must not flow into a central data store to be processed there by a workflow engine. In the context of DIVA Iteration 5, this topic is addressed in detail (R5.1). A flexible data analysis network concept was developed and published in [Teb20, Paper III.]. The data analysis network presented allows users to configure workflows according to their needs.

6.3 DP3: Principle of Interoperability

Table 6.3: DP3: Principle of Interoperability

Design principle title	**DP3: Principle of Interoperability**
Aim, implementer, and user	For the data catalog to seamlessly cooperate with other data catalogs and other systems in general (aim).
Mechanism	Standardize, document, and publish the metadata models, APIs, processes, and source code.
Rationale	Data catalogs must be able to embed themselves in existing technology landscapes and understand many different systems to be used reasonably.
Extracted from	R1.2, R5.1, R8.1
Supported by	[Ade17; Alv22; Ate14; Bar22b; Bog21; Bor22; Cap17; Cla17; Cyg10; Di16; Dih15; Ehr21a; Hey15; Jef14; Joh90; Kir23; Kir19a; Klí18; Kře19; Lee16; Oli16a; Pes15; Rya22; Sch17; Sch18b; Scr22; Tor19; Tyg16; Tzi21; Wan17a; Wan16; Yu16]
Design Features	• DF3.1: Standardized Metadata Models • DF3.2: Standardized API Documentations • DF3.3: Open Source

This section discusses the third DP for data catalogs. Table 6.3 shows an overview of the DP. "Interoperability is the ability of two or more software components to cooperate despite differences in language, interface, and execution platform." [Weg96, p. 285]. On the one hand, this is important to enable seamless integration and cooperation of new systems in an existing system landscape. If this interoperability would not exist and each system would have to be adapted individually, this would quickly lead to a combinatorial explosion. When each system must be able to talk to every other system, $\binom{n}{2}$ connections

must be handled. With five systems, there are ten connections; with ten systems, 45 connections, and 100 systems already have 4950 connections. It makes sense to strive for interoperability if the system under consideration is just one part of a larger set. On the other hand, interoperability is also important "[...] to build coherent services for users, from components that are technically different and managed by different organizations." [Arm02]. This way, users are not burdened with familiarizing themselves with systems that do the same or similar things at their core.

Interoperability can have different levels. Wang et al. describe five levels of interoperability [Wan09]. These range from isolated systems to standard conceptual models and semantic consistency (see Figure 6.2). The further one progresses in the level of interoperability, the less complex and, thus, the more effective the system connection becomes.

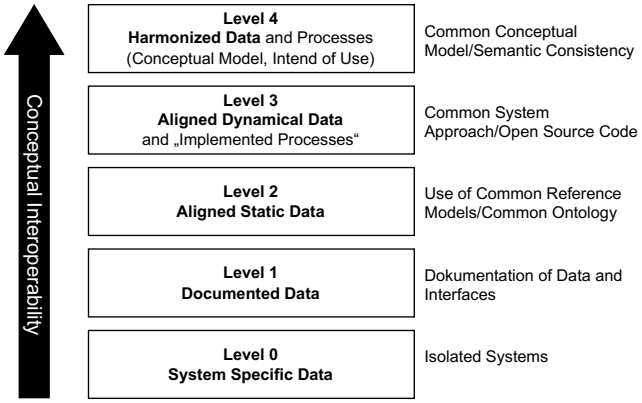


Figure 6.2: Levels of conceptual interoperability by Wang et al. [Wan09]

In order to achieve the goals mentioned above and a high level of interoperability, technical, content, and organizational agreements must be made [Arm02]. Technical agreements deal with the use of uniform data formats or communication protocols. Content agreements try to create a standard about the content to be exchanged and its interpretation. Organizational agreements involve life cycle management, access rights, and payment modalities. Although reaching an agreement with many parties is challenging, it is necessary to some extent to ensure a seamless integration that also benefits users.

Interoperability is an essential topic for data catalogs because a lack of standardization can lead to implementation failures [Lee16]. In practice, the lack of measures to increase interoperability in data catalogs is particularly noticeable [Ehr21a]. Data catalogs rarely represent an isolated system (see Level 0 in Figure 6.2), as this would make it difficult for them to fulfill their task. They embed themselves in existing technical landscapes and must be able to interact with other systems to perform their tasks effectively. For example, data catalogs must be able to communicate with other data catalogs or applications to exchange data offerings through harvesting [Fri14; Maa10a; Mar13]. The aim is to create metadata standards for this purpose [Joh90]. One of the best-known standards in the field of data catalogs, mainly used by ODPs, is DCAT. Many works use this standard [Alv22; Hey15; Kir19a; Klí18; Wan16], want to improve it [Bor22], build on it [Cap17; Kir23; Rya22; Scr22;

Yu16] or explore its use in practice [Bar22b]. Such a standard makes it possible, for example, to create a unified view across many data catalogs [Oli16a]. Also, apart from DCAT, there are attempts to create standards for domain-specific metadata in data catalogs [Ate14; Dih15; Jef14; Kře19], life cycle standards [Teb23d, Paper VII.], and standards for entire domain specific catalogs [Cla17; Pes15]. In addition to standardization of metadata models, standardization of the data catalog and making it available as free open source software is also an issue [Tor19]. "[...] efficient code reuse among public organizations [...] would allow not only better use of taxpayers' money but leverage network effects toward incremental and cumulative innovation." [Lee16]. CKAN, for example, is a well-known and freely available data catalog that is primarily used in the field of ODPs [Sch17; Sch18b; Tzi21]. Overall, one tries to implement linked data principles to increase interoperability [Ade17; Cyg10; Di16]. Also, attempts are being made to establish interoperability in data catalogs by standardizing keywords [Bog21; Tyg16] and identifiers [Wan17a].

The data catalog community should strive for a high degree of interoperability. At the very least, level 3, according to Wang et al., should be implemented (see Figure 6.2). This level states that in addition to static interoperability through uniform protocols and data models, the internals of the data catalog are also published and standardized. This is necessary to understand the data catalog's behavior during integration. For example, incorrect behavior can be traced more easily. In the simplest case, the entire data catalog is published as FOSS, allowing third parties to gain insight.

6.3.1 DF3.1: Standardized Metadata Models

To implement DP3, standardized metadata models should be used.

The standardization of data models plays a vital role in many domains in which data must be exchanged between systems. It should be emphasized that this is particularly important when the systems are located in different areas of responsibility. An example is data exchange in the healthcare sector, where health data or clinical data must be exchanged between doctors, hospitals, and health insurance companies [Gam21; Hus15]. In construction, creating a standardized data model is also important so that the various industries on construction sites can all access project information [Bor18]. The same logic applies to data catalogs. As soon as the data catalog needs to exchange data with other data catalogs or other systems that want to further process the metadata of the data catalog, a standardized data model should be used.

Implementation Recommendations

For the concrete implementation in data catalogs, it is advisable to build the foundation on DCAT, which acts as the lingua franca of the data catalogs. Especially in the area of ODPs, DCAT [Alb23] has emerged as a standard. DCAT is a Resource Description Framework (RDF) vocabulary and is intended to support that the inventoried datasets in the ODPs can be exchanged more easily among each other. Currently, DCAT is available in version 3.0[1]. Building on DCAT, extensions can be created to meet one's requirements.

1 `https://www.w3.org/TR/vocab-dcat-3/` [Accessed: June 23, 2023]

Examples of this can be found abundantly [Cap17; Rya22; Scr22; Yu16]. We, too, have developed a metadata model based on DCAT, which describes data in terms of an economic good [Spi18, Paper I.].

Also, when storing further information about the data, it should always be checked whether a widely used standard can be used. For example, ODRL can describe who can perform which actions on the data [Vos19]. Likewise, we refer to our work, establishing a standard to describe the end of the data life cycle [Teb23d, Paper VII.].

It should also be noted that no obsolete standards are used. An example is the so-called *Information Model*, used in the context of the IDS to exchange data offers between the data space data catalogs. It was created due to limitations in the first version of DCAT. However, the creators already note in their publication that "many of these limitations have recently been addressed with DCAT 2 [...]" [Bad20, p. 14].

Application Examples

In practice, a plugin for the widely used CKAN adds support for DCAT.[1] Many ODPs are based on CKAN. An example of a data catalog built on CKAN is the EU data portal. Here, the DCAT-AP[2], a special extension for describing public sector datasets is used [Kir19b]. DCAT is also used in the context of the EDC. The EDC Federated Catalog uses DCAT to exchange data offerings.[3]

In the DIVA Iteration 1 the data model M4DG (R1.2) was developed and published [Spi18, Paper I.]. This model is based on DCAT and extends it with properties to describe data in terms of an economic good. Since version 1.0, DIVA implements the M4DG data model and is thus able to answer queries in the DCAT standard as well. In Iteration 8, the Destroy Claim data model (R8.1) was developed and published to describe the end of the data life cycle [Teb23d, Paper VII.]. DIVA implements the JSON version of the model.[4]

6.3.2 DF3.2: Standardized API Documentations

To implement DP3, standardized API documentation should be provided.

Data catalogs are a central point of contact for answering questions about available data. It can be necessary for external systems to obtain information from the data catalog. A description of the available API should be provided to enable the systems to do this. The description should be as comprehensive as possible in order to accelerate integration.

Implementation Recommendations

Many data catalogs are accessible via HTTP, which is why it makes sense also to provide and document the HTTP API. Using existing standards for describing HTTP APIs is

1 `https://extensions.ckan.org/extension/dcat/` [Accessed: June 23, 2023]

2 `https://joinup.ec.europa.eu/collection/semantic-interoperability-community-semic/solution/dcat-application-profile-data-portals-europe/release/211` [Accessed: June 23, 2023]

3 `https://github.com/eclipse-edc/Connector/tree/f8db7524055ae2fd98e1b72efb9ff26fecddcf28/docs/developer/architecture/ids-dataspace-protocol` [Accessed: June 23, 2023]

4 `https://github.com/DaTebe/destroyclaims/blob/9a8a6e5e432312175cc439f333c45620924400fc/docs/destroy-claim.md` [Accessed: June 23, 2023]

recommended. One widely used standard is OpenAPI[1]. OpenAPI can be used to describe HTTP APIs using JSON or YAML. Another option, if the data catalog is accessible via HTTP, is GraphQL[2]. GraphQL provides a solution that combines the API description, data model, and query execution. However, this solution can only be used if the programming language used within the data catalog supports GraphQL.

In addition to synchronous APIs such as HTTP, asynchronous APIs can also be used in a data catalog. Examples include message queuing systems such as RabbitMQ[3] or Apache Kafka[4]. Also, WebSockets[5] represent an asynchronous communication option that a data catalog can implement. AsyncApi[6] represent a way to describe what channels exist and how messages are structured.

Application Examples

The EU data portal provides an OpenAPI specification for searching the catalog.[7] For visualization, redocly[8] is used (see Principle of Visualization). OpenMetadata also provides an OpenAPI specification to describe its API.[9]

DIVA uses OpenAPI with Swagger-UI[10] for visualization. DIVA also provides an AsyncAPI specification for Kafka so that external systems can read and process events.[11]

6.3.3 DF3.3: Open Source

> *To implement DP3, make the data catalog available as open-source software and use established open-source software*

As Wang et al. describe in their work, it is not always sufficient to create documentation and to use uniform models and interfaces. Internal processes in the software must also be visible in order to be able to understand how results are composed [Wan09]. This visibility is crucial for data catalogs to exchange metadata with other data catalogs or receive data from third parties. As a simple example, we can consider a quality metric that reflects the accuracy of a dataset. Two different implementations in two different data catalogs may produce different results despite having the same data set. Even though there is a standard model describing what this quality metric means and expresses, there is still room for interpretation. To address this, the code that calculates the quality metric must be freely available to compare implementations and make adjustments as needed. Data catalogs

1 `https://spec.openapis.org/oas/v3.1.0` [Accessed: June 23, 2023]
2 `https://graphql.org/` [Accessed: June 23, 2023]
3 `https://www.rabbitmq.com/` [Accessed: June 23, 2023]
4 `https://kafka.apache.org/` [Accessed: June 23, 2023]
5 `https://datatracker.ietf.org/doc/html/rfc6455` [Accessed: June 23, 2023]
6 `https://www.asyncapi.com/` [Accessed: June 23, 2023]
7 `https://data.europa.eu/api/hub/search/` [Accessed: June 23, 2023]
8 `https://redocly.com/redoc/` [Accessed: June 23, 2023]
9 `https://docs.open-metadata.org/v1.3.x/developers/architecture/code-layout#api` [Accessed: May 13, 2024]
10 `https://swagger.io/tools/swagger-ui/` [Accessed: June 23, 2023]
11 `https://github.com/FraunhoferISST/diva/tree/0c5e4cbed2ca197e77bb41d7ac875fbe4b655b61/core/services/entity-management/defaultEntities/asyncapi` [Accessed: June 23, 2023]

can only collaborate securely in a network and exchange data offerings when this type of interoperability is also provided. Otherwise, there is always the risk that all interfaces and models will be complied with, but there will be deviations in content.

Implementation Recommendations

Using platforms with a wide distribution is recommended for the provision of code. This includes, for example, GitHub[1]. Third parties can look at the code, check it out, and try it themselves. Code hosted on GitHub can also be archived for the long term using Zenodo[2]. This ensures that the code can still be accessed in a few years to understand how specific algorithms work.

Executable code can also be provided as container images to increase interoperability. This allows third parties to deploy code by using, for example, Docker[3] or Podman[4]. However, it should be noted that the source code may not be easily extractable from the image, making traceability difficult.

Application Examples

Since September 2021, DIVA is available in version 3.0 as FOSS under the Apache 2.0 license on GitHub [Teb23b]. The FaaS responsible for data profiling can also be viewed here. Thus, other data catalog providers can understand how DIVA analyzes inventoried data and automatically extracts and generates the metadata. An example of this is the detection of keywords from texts, which was implemented as part of DIVA Iteration 6 (R6.1). Other data catalogs can use the source code to adopt the implementation if required. This feature is essential if one wants to outsource the data profiling to other systems to use available resources efficiently or to fulfill data sovereignty requirements [Teb20, Paper III.].

In the DIVA Iteration 5, we designed a system architecture to distribute data profiling tasks to arbitrary machines in a peer-to-peer network (R5.1). The individual tasks were made available as Docker images to allow the machines to execute the tasks. The machines can then download and cache these images. This feature ensures high interoperability in a network of computers that do not know each other.

The data catalogs DataHub, OpenMetadata, CKAN, and EDC Federated Catalog are also available as open-source software.

6.4 DP4: Principle of Context

This section discusses the fourth DP for data catalogs. Table 6.4 shows an overview of the DP. Data is rarely a stand-alone entity and must be placed in a larger context for a more comprehensive understanding. Depending on the situation, a wide variety of contextual information is required. For example, in one use case, it may be of interest where and by whom the data was collected, e.g., to make implications about data quality. In another use case, knowledge about the data's technical format is required to evaluate whether it

1 `https://github.com/` [Accessed: June 30, 2023]
2 `https://zenodo.org/` [Accessed: June 30, 2023]
3 `https://www.docker.com/` [Accessed: June 30, 2023]
4 `https://podman.io/` [Accessed: June 30, 2023]

Table 6.4: DP4: Principle of Context

Design principle title	**DP4: Principle of Context**
Aim, implementer, and user	For the users of the data catalog (user) to gain a better understanding and deeper insight into the inventoried data (aim).
Mechanism	Provide data-system relationships, data lineage, data linkage, business context, and technical context by enriching metadata, connecting data to related entities in the data catalog, and answering questions regarding the what, who, where, when, why, and how.
Rationale	Context allows for faster, better, and more accessible data understanding.
Extracted from	R1.1, R1.2, R6.1, R7.2, R7.3, R7.5
Supported by	[Ade17; Ber22; Cho20; Dos19; Ehr21a; Hal16; Jia19; Lab20a; Lac17; Lee12; Mil15; Neu18a; Ojo20; Por20; Sen04a; Sen18; Sha16b; Sha05; Ško19; Sko19]
Design Features	• DF4.1: Metadata With Contextual Information • DF4.2: Data Networks

needs to be transformed before it can be used. Generally speaking, context helps us better understand the data and make more informed decisions about using it. For example, sensor data can only be meaningfully analyzed and interpreted with knowledge of the type of sensor, its measurement accuracy and intervals, and its physical location. Data does not necessarily carry the context in which it was generated or used. Shannon et al. summarize the consequences of missing or inaccessible contextual information about data as follows: "Unfortunately, if word-of-mouth has insufficiently propagated the information about the data ownership and access procedures, researchers may waste time and effort creating a new dataset, use a dataset inappropriate for the problem at hand, or worst of all, abandon the project." [Sha05, p. 98]. Thus, a solution is needed that provides users of the data with the contextual information they need.

Data catalogs represent a solution to collect contextual information and make it available to users. Data catalogs should be used to "[...] not only enable users to find and identify data, but to provide all necessary contextual information so that users can understand them [...]" [Lab20a, p. 202]. Depending on the data type, a data catalog must be able to store different types of contextual information. As an absolute baseline, we need to fill in the different types of metadata according to Riley (see Section 2.1.1). In detail, Shanmugam and Seshadri designed categories for the different types of context information that should be stored in a data catalog. These include *data system relationships*, *data lineage*, *data linkage*, *business context*, and *technical context* [Sha16c]. In particular, the

modeling or exploitation of relations between data in data catalogs in the form of, for example, similarities, lineage, provenance, or knowledge has generated much work [Ade17; Ber22; Cho20; Dos19; Hal16; Lac17; Mil15; Neu18a; Ojo20; Por20; Sen04a; Sen18; Ško19; Sko19]. Also, work has been published that considers tags crucial contextual information in data catalogs [Jia19; Lee12]. Ehrlinger et al., similar to Shanmugam and Seshadri, were able to identify the provision of business context as an essential building block in a data catalog [Ehr21a]. In addition, Ehrlinger et al. also identified information about responsibilities as important. Assigning responsibilities clearly defines who is responsible for the quality and reliability of the data. This can also increase trust in the data. Depending on the context, different people may be responsible for different aspects. For example, one person may be responsible for metadata maintenance and another for data content maintenance. A company should define specialized roles for different responsibilities, such as data architect, data owner, or data steward [Lab20a].

6.4.1 DF4.1: Metadata With Contextual Information

> *To implement DP4, extend the metadata model by as many context-related attributes as possible.*

A lot of context information can be represented as simple values. Therefore, it makes sense to provide corresponding attributes in the metadata model of the data catalog. If we look again at the example with the sensor, we can save the data format, the encoding, the used separators, and the generation interval as *technical context*. In the scope of the *business context*, one could store, for example, where the data is used in the company and who is responsible. *Data-system relationships*, *data lineage*, and *data linkage* can also be partially represented in this way. However, we recommend the use of a data network (see DF 4.2).

Implementation Recommendations

We recommend that the data model can be adapted if necessary (see Principle of Flexibility). However, domain experts should only do this to avoid uncontrolled growth of the model. As far as possible, standards should be used to maintain interoperability (see Principle of Interoperability). Ideally, context information is determined automatically as part of the data profiling (see Principle of Automation). Where applicable, context knowledge should be visualized (see Principle of Visualization). An example of this would be the display of the position of a data-collecting sensor on a map.

Application Examples

In DIVA Iteration 1, contextual properties beyond the scope of DCAT have been introduced to DIVA (R1.2). This includes information about responsibilities and where the data is used in the organization. Further properties are more detailed information about technical aspects, e.g., whether the data is compressed or encrypted. The attributes are part of the M4DG model [Spi18, Paper I.]. In DIVA Iteration 7, DIVA has been extended to add arbitrary additional context attributes (R7.2). Another example is DataHub, where users

can add business attributes to datasets to improve understanding and searchability.[1]

6.4.2 DF4.2: Data Networks

> *To implement DP4, establish relationships between the data and other entities stored in the data catalog, such as other data, users, publishers, or projects, creating a data network.*

Data can have relationships with other data or entities that provide insight into context. Relationships can exist, for example, between similar data, the person responsible and their data, or the data and its publishers. A graph is built implicitly or explicitly over these relationships, depending on the persistence technology used. Thus, it is straightforward to map *Data System relationships*, *Data Lineage*, and *Data Linkage*. The resulting graph is called a data network in the context of this thesis.

Implementation Recommendations

The data network should contain as many relevant relations as possible. Many relations and a high amount of data can result in complex graphs. It is recommended to persist the relations in a native graph database [Cho20]. These also allow for deep graph searches and can execute typical graph algorithms, which can be interesting, e.g., for clustering or shortest path tasks. The user should be able to visualize the data network and interact with it (see DF 6.2). The majority of relationships should be able to be stored in the data network by an automatism (see DF 1.2). Many data catalogs in the open data sector try to implement linked data principles and, thus, semantic mapping technologies. A graph structure is also built, and relations can be viewed here. It is necessary to inform oneself about the respective solution's advantages and disadvantages and make a suitable decision in one's own context. An example is the data catalog of NASA, which neither uses semantic mapping nor graph databases but is based on cognitive search engines [Bug22].

Application Examples

Since DIVA Iteration 1, DIVA can model the relationships between data and entities on a small scale. For example, it is possible to represent whether alternative data sets exist and which processes and applications use them. Figure A.9 shows on the left how the relations are represented in the DIVA 1.0 web interface. Users can then use the information to assess the data (R1.1). In DIVA Iteration 4, modeling relationships was implemented more generically in the form of groups (R4.3). This way, users can create groups themselves and thus create relations between data.

In DIVA Iteration 6, keywords were automatically extracted from medical documents (R6.1). These were then used in DIVA Iteration 7 to link documents with similar keywords (R7.5). This allows DIVA users to see at a glance if there is any other relevant data for them. The result is then visualized in the DIVA web application (see Principle of Visualization).

With DIVA 4.0 and as part of DIVA Iteration 7, a more comprehensive data network has been implemented (R7.3). The network is persisted in the Neo4j graph database. This

1 `https://datahubproject.io/docs/businessattributes` [Accessed: May 13, 2024]

allows DIVA to flexibly create relations between entities and enrich them with additional information such as labels, weights, or other values. Users can view the links to other data, projects, and further entities via the web interface and traverse through the graph (see Figure A.18). It is intended to support an explorative search for suitable data in the existing data network.

6.5 DP5: Principle of Data Life Cycle Management

Table 6.5: DP5: Principle of Data Life Cycle Management

Design principle title	**DP5: Principle of Data Life Cycle Management**
Aim, implementer, and user	For the users of the data catalog (user) to share and gain necessary operational knowledge about the data (aim).
Mechanism	Equip the data catalog with data life cycle information and associated management functions.
Rationale	The availability of data life cycle information enables users of the data catalog to assess operational consequences immediately.
Extracted from	R5.2, R8.1
Supported by	[Dag16; Kim21; Qui20; Rya22; Sen04a]
Design Features	• DF5.1: Data Usage Policies • DF5.2: End-of-Life Data Management

This section discusses the fifth DP for data catalogs. Table 6.5 shows an overview of the DP. Consideration and attention to the data life cycle is vital to working with data. The life cycle of data can be divided into different phases. In literature, many data life cycle models have been developed for different applications and domains in the last ten years alone [Cha15a; Dem14; Dem20; Fau13; Hub13; Kha14; Kow17; Len14; Lin14; Ma14; Mia17; Mic15; Pat16; Wis16]. Typically, one finds phases such as creating, storing, processing, using, and deleting data. Having access to information about these phases can have significant advantages. Better and more informed decisions can be made when it is known how the data was collected, how it is allowed to be processed, and when it may become obsolete and need to be deleted. This knowledge can also establish enterprise-wide compliance by knowing, for example, that data contains personally identifiable information and may not be used for purposes other than those specified in the contracts. One problem is that a single entity rarely exercises control over the entire data life cycle. Data is distributed through different channels to different stakeholders. In doing so, the data is often subject to changes that negatively affect its quality or is applied in contexts where it should not be applied [Hub13].

One solution might be to use a data catalog as a central point of contact for questions about the data life cycles. “Data catalogues have become an important pillar in the data

management lifecycle. Indeed, almost every step of the data lifecycle is described in the metadata fields or accessible through the data catalogue online interface." [Qui20, p.141]. A data catalog should already inventory all relevant data. Thus, the catalog can also provide insights into the available data and information about the data life cycles [Sen04a]. Interested parties can see how to use the data and whether it must be deleted after a certain period. Data catalogs can use specialized data models for this purpose [Teb23d, Paper VII.], [Rya22], or also rely on standards such as ODRL [Dag16] (see DF 5.1). Data life cycle management refers to the inventoried data and can be extended to the metadata in the data catalog. Even in a closed system like CKAN, where one can store metadata as well as data, incorrect management can result in, for example, data being deleted but metadata being unintentionally left in the data catalog [Kim21]. It makes sense, thus, to hold data life cycle information for data and metadata. If other instances do not already perform this task, managing the data life cycle directly in the data catalog makes sense. Users must have access to robust modeling options and receive adequate support throughout the modeling process.

When considering the data life cycle, it is primarily a matter of operational knowledge. For example, who may do what with which data, when and where it may flow, and when it must be removed. This perspective also distinguishes this DP from the Principle of Context.

6.5.1 DF5.1: Data Usage Policies

> *To implement DP5, provide usage policies for the inventoried data in the data catalog to prevent unauthorized access, mishandling or to ensure legal and regulatory compliance.*

For the users of a data catalog, it is of interest whether there are conditions that must be met when using the data. These conditions can vary greatly depending on the context. For example, restrictions can be placed on the context in which the data can be used, who may access it, or whether it may be copied and adapted. Having this information attached directly to the entries in the data catalog makes it easier for users to decide whether they can use the data for their use cases.

Implementation Recommendations

Using existing standards for formulating usage policies is a good idea during implementation. One widely used standard is ODRL (see DF 3.1). This standard describes possible or forbidden actions regarding digital rights management on data. If the DCAT standard is used, it is even more appropriate to use ODRL since it is directly supported.[1]

Application Examples

In DIVA Iteration 5, a connection of DIVA to IDS compliant data spaces was implemented. A DSC was implemented as the basis for offering the data inventoried in the data catalog within the scope of the IDS. Usage policies can be attached to the data, which must be

1 `https://www.w3.org/TR/vocab-dcat-3/#Property:resource_has_policy`

adhered to by other participants in the IDS. The usage policies were represented using the IDS Usage Control Language, which is an adaptation of ODRL and was used in the context of data space data catalogs [Eit21]. In the meantime, however, IDS also uses DCAT and thus ODRL policies.[1] The EDC Federated Catalog also stores usage policies in the form of ODRL policies.

6.5.2 DF5.2: End-of-Life Data Management

> *To implement DP5, enable end-of-life data management in one's data catalog to, among other objectives, ensure regulatory and legal compliance, prevent the proliferation of poor-quality data, protect security-critical data, save costs, or destroy broken data.*

Several current and future challenges reinforce the urgency to deal intensively with the end of the data life cycle and enable effective management. First, regulatory requirements such as laws and standards make effective end-of-life data management necessary. One example is GDPR, which describes the right to be forgotten. For example, data must be deleted if "[...] the personal data are no longer necessary in relation to the purposes for which they were collected or otherwise processed;" [Eur16]. Also, if hardware in the form of hard drives, USB sticks, or the like must be disposed of or sold, it should be ensured that standards such as NIST 800-88 [Reg15] are applied for the secure deletion of the data. This procedure can prevent sensitive data from leaking to third parties. Another argument favoring end-of-life data management is that most data is only used once and remains stored on various systems [Tra18]. On the one hand, this generates unnecessary costs. While in past years in the cloud area, customers were bound by low prices, current price increases[2] can be a reason to part with unused data. Environmental considerations can also be an incentive to part with unused data. This fact is shown by current research in the field of digital decarbonization [Jac22]. Also, the European energy crisis [Cou22] gives reason to become active here. In addition, deleting low-quality data is essential to prevent unwanted dissemination within a company and avoid possible negative effects on processes and products [Cha14]. Deleting data can also be an important aspect in the area of data ecosystems and data spaces [Jun22], for example, when one wants to inform participants to delete certain data because it contains errors or because the usage time has expired. Finally, there are technical factors to consider. Deleting corrupt, invalid, or disruptive data and the need for efficient calculations by removing unnecessary data may require deleting data [Gao19; Pac18; Rea12].

Despite this multitude of reasons and an ever-increasing mass of data to consider, little attention is paid to deleting data and managing the end of the data life cycle [Teb21, Paper IV.], [Teb23a, Paper V.], [Teb23c, Paper VI.]. A data catalog is a suitable solution to support the management of the end of the data life cycle and to communicate its need to its users [Teb23d, Paper VII.]. A data catalog should already have inventoried all relevant

1 `https://github.com/International-Data-Spaces-Association/ids-specification/blob/ff019e4f12a42c10dcc20640ee9a525108460230/catalog/catalog.protocol.md`

2 `https://cloud.google.com/storage/pricing-announce` [Accessed: June 30, 2023]

data. Therefore, it is a good idea to offer users information about the end of the data life cycle in the data catalog. Also, the persons responsible for the data should be enabled, where meaningful, to model the end of the data life cycle directly within the data catalog.

Implementation Recommendations

A standardized model is recommended to describe the end of the data life cycle. On the one hand, this guarantees that the usual scenarios can be represented. On the other hand, different instances can achieve a uniform understanding, exchange the model among each other, and interpret it uniformly. This communication ability is crucial for automation (see Principle of Automation) and interoperability (see Principle of Interoperability). Thus, different actors can remove data that has reached the end of its life cycle from the data landscape.

ODRL offers the option of end-of-life modeling. Deletion is described as an action on data and can be controlled by conditions. However, ODRL was not explicitly designed for modeling the end of the data life cycle. Destroy Claims represent a solution developed precisely for this situation [Teb23d, Paper VII.]. They can be derived in various data formats such as JSON and thus be integrated into arbitrary technologies. In contrast to ODRL, specific aspects of the end of the data life cycle are emphasized here. For example, different end-of-life scenarios can be modeled for different situations. For these scenarios, responsibilities can then be modeled on a fine-granular basis. Thus, it is possible to model who is responsible for the data, who is responsible for which part of the modeling, and who is responsible for the overall claim. Depending on the scenario, different standardized reasons for deletion can be assigned to trigger automated effects, such as notifying the data protection officer.

In addition to the use of standards, the data catalog should support its users with suitable automation (see Principle of Automation) and visualizations (see Principle of Visualization). On the one hand, since this is a complex topic, the user should be supported in the modeling process. Support here means that the user can choose from pre-built end-of-life modeling for certain types of data. Ideally, a data catalog can also automatically apply suitable end-of-life models for certain data types. Furthermore, the catalog should explain the impact of the model to the user. This explanation can be done through suitable visualizations that, for example, highlight the affected parts of a system. In general, however, it can be stated that best practices in this area have yet to be developed [Teb23d, Paper VII.].

Application Examples

As part of Iteration 8, end-of-life data management was implemented in DIVA 4.1 (R8.1). The Destroy Claim model, which can describe the end of the data life cycle, serves as a basis here [Teb23d, Paper VII.]. Users can create their own Destroy Claims through the user interface to model the end of the data life cycle for inventoried data. An example view of the DIVA web interface is provided in Figures A.22, A.23, and A.24. Further details can be found in [Teb23d, Paper VII.].

Table 6.6: DP6: Principle of Visualization

Design principle title	**DP6: Principle of Visualization**
Aim, implementer, and user	To allow the data catalog user (user) to identify relationships and patterns regarding data and metadata more easily and quickly (aim).
Mechanism	Provide visualizations of non-trivial aspects and tailor them to fit different use cases and roles.
Rationale	Humans can recognize connections and patterns better and faster when visualizing data. Data catalogs contain a lot of metadata that a user needs to process.
Extracted from	R1.1, R1.3, R6.2, R7.4
Supported by	[Ben14; Car15; Hol19; Kop21; Nik21; Pie18; Roz12; Tim23; Zui16]
Design Features	• DF6.1: Metadata Visualization • DF6.2: Data Network Visualization

6.6 DP6: Principle of Visualization

This section discusses the sixth DP for data catalogs. Table 6.6 shows an overview of the DP. Visualizing data is vital in many technical systems [Teg99]. The more complex the data that needs to be analyzed and interpreted, the more likely visualizations can be an asset or even an unavoidable necessity. Primarily, this is because humans are inherently capable of recognizing and processing visual patterns. It is also exploited in the visualization of data [Apa15]. Thus, the user can make decisions based on data more easily [Bur19; Lur07]. This advantage can be observed especially with complex data [Bas16]. When selecting the visualization, it is crucial to ensure that it fits the underlying data as well as possible. Many publications deal with processing different types of visualizations and assigning them to suitable use cases [Ste10; Tel14]. Also, state-of-the-art compilations were elaborated [Ano20; Pos02]. In addition to selecting the appropriate visualization, aesthetics also plays a role that should be considered. There are indicators that aesthetic visualizations are evaluated faster and with fewer errors [Caw07].

With their large amount of stored metadata, data catalogs represent a class of systems that benefit from visualizations [Zui16]. The amount of metadata and complexity is not trivial, so users must be supported here in analysis, evaluation, and decision-making [Car15]. Visualizations can be used, for example, for quality assurance or to investigate the data catalog in an exploratory way to make discoveries [Tim23]. Work in data catalogs also deals with using different types of charts [Pie18] or maps [Kop21]. Also, topics like the visual summary of data offerings of a data catalog [Ben14] or the representation of data lineage have already been addressed in the literature [Hol19]. Despite the importance of visualizations, these are often poorly implemented, for example, in the area of ODPs [Nik21].

One way to counter this could be by using frameworks that specify the user interfaces of data catalogs [Roz12].

6.6.1 DF6.1: Metadata Visualization

To implement DP6, provide visualizations for non-trivial metadata.

Metadata is a crucial way to gain better insight into the inventoried data of a data catalog. A data catalog usually includes descriptive, administrative, and structural metadata. They can exist in various statistical analyses, metrics, indicators, and other types. The more complex the metadata, the more difficult it becomes to interpret a purely textual representation. Therefore, metadata should be visualized for users when it is no longer trivial.

Implementation Recommendations

In principle, the visualizations should integrate visually into the look and feel of the data catalog. It is recommended that widely used libraries be used for visualization. On the one hand, getting as far as possible all types of required visualizations from one source benefits the aesthetics [Caw07]. On the other hand, the probability of getting support for a longer time is higher. Users should also be able to interact with the visualization. It should be possible, at least partially, to allow select, explore, reconfigure, encode, abstract/elaborate, filter, and connect actions [Yi07].

Another recommendation is to implement dashboards where metadata must be displayed cohesively, especially if it spans the data catalog globally. "A dashboard is a visual display of the most important information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance." [Few07, p. 1]. Evaluations such as trends or key indicators often flow into these centrally and complement each other. It is designed so that users can easily recognize correlations and patterns across different data sources. This is done, for example, through visualization using charts, graphs, gauges, and maps. Their potential centralization also makes them suitable for collaborative work tasks. Dashboards are already used in many areas. These include for example, systems from medicine [Sta16; Wil14] or from teaching [Jiv18; Ver13], where the use of dashboards has increased the efficiency of work. In data catalogs, dashboards can be an important component that can help users evaluate their own data offerings or make decisions. They should be customizable by users to their own needs and should also display differently processed data depending on the user's role (see also Principle of Flexibility). Finally, real-time dashboards should be mentioned. These allow the user to gain insight into the available data in real-time or near real-time.

Application Examples

In DIVA, the visualization of metadata for individual inventoried data was introduced in DIVA Iteration 6 and has been a permanent feature since version 3.0. The focus is, in particular, on the visualization of statistical analyses of tabular data. For example, analysis of the frequency distribution or the average of the values is done. Figure A.13 shows this in DIVA 3.0. Figure A.19 gives an insight into the visualization in version 4.0.

The visualizations are implemented using the chart.js[1] library, which offers a wide range of visualizations and adapts to the existing look and feel of DIVA.

Due to project requirements, dashboards have also been implemented in DIVA. DIVA has implemented a management dashboard since DIVA Iteration 1 that displays overarching key metrics (see Figure A.7) (R1.3). The dashboard has continued to be customized over the development iterations, as users have found it a helpful tool. It is also in the latest version (see Figure A.20).

6.6.2 DF6.2: Data Network Visualization

To implement DP6, visualize the relations between data and other entities stored in the data catalog.

In data catalogs, a data network includes knowledge about how data relates to other data, projects, people, and entities. It can also include from which other data catalogs the data was imported or from which vendor it originated. It can be said that the data network can contain information about data-system relationships, data lineage, and data linkage [Sha16c]. The network can provide an interesting insight into the data for the users of the data catalog. A visualization of the data network allows exploratory searches for interesting relations. For example, users can see if the data they used was also used in other projects; thus, synergy effects are possible. Also, patterns and clusters can be discovered more easily.

Data flows, e.g., in the context of data lineage, can also be mapped in data networks and should be visualized. Data lineage describes the flow of data and its stations during the life cycle. This description includes where the data comes from, which intermediate stations it passes through, which transformations it has undergone, and where it is finally stored. This feature is essential because the data flows are no longer trivially manageable. The systems holding and processing the data can be distributed worldwide and involved in vast quantities. Traceability is important, however, in order to be able to make statements about possible quality problems or errors [Woo97], but also, in particular, for documentation and audit purposes in order to be able to trace whether the data comply with regulatory requirements such as laws or other compliance rules [Fre21]. Data catalogs should, therefore, address data lineage [Jah22; Sha16c]. Since the data flow can be very complex, it should be visualized.

Implementation Recommendations

In order to analyze the data with different objectives, the visualization must be customizable. In addition to data-related filters, customization options should also influence how the network is rendered. It should be kept in mind that users are not technical experts. Customizability should be made as easy as possible. Orientation for the implementation can be provided by existing tools like Gephi [Bas09]. Shared best practices should also be consulted in drawing [Di 94; Her00] and visualizing [Bör03] networks and graphs. Care

1 `https://www.chartjs.org/` [Accessed: May 31, 2023]

should be taken to ensure the visualization libraries can handle different data formats to simplify integration.

Application Examples

DIVA had already implemented a simple visualization for certain relations in version 1.0 (see the left side of Figure A.9). The implementation resulted from the requirements in DIVA Iteration 1. It was a building block to make it easier for users to assess the data in the data catalog. However, the visualization was not customizable. The feature was then not pursued in further development iterations for the time being. With version 3.0, we implemented a new variant (see Figure A.14). There, grouped data and related entities could be displayed together in one view (R6.2). In DIVA Iteration 7, a new approach to data networks was implemented (R7.4). Users can navigate the network by interacting with the nodes and edges and have subgraphs loaded on demand (see Figure A.18). The edges in the graph are labeled and can also contain additional information. DIVA 4.0 stores the data network and further data, such as similarity scores in the graph database Neo4j. A native graph structure at the persistence level makes it easier to perform queries. The application of typical graph algorithms is also facilitated and scales better. Finally, the visualization also benefits from this technical realization. Often, query results from widely used database solutions can be imported directly or with little effort into visualization libraries. In DIVA, the query results are rendered using the vis.js[1] library.

Although the provision of data lineage knowledge and its visualization is undoubtedly an important component of data catalogs, this has never been addressed in the DIVA. An impression of how to implement such visualization can be taken from OpenMetadata and DataHub.[2, 3]

6.7 DP7: Principle of Data Assessment

This section discusses the seventh DP for data catalogs. Table 6.7 shows an overview of the DP. Data assessment is essential to identifying the appropriate data for one's project. Many aspects can play a role here. One widely used aspect is the assessment of data quality. Data quality is generally described as "fitness for use" and is important when working with data [Wan96]. Data is often used in decision-making and should, therefore, be of high quality in various dimensions such as believability, accuracy, objectivity, completeness, or reputation [Wan96]. In companies, the data quality directly influences the overall performance and affects things like decision-making, product quality, and project costs [Cha14]. Also, poor data quality can lead to dissatisfied customers, employees, and general distrust in the company [Red98]. Therefore, it is unsurprising that quality also plays an important role in selecting suitable data for one's project [Teb23a, Paper V.].

Another aspect is risk assessment. The risk that valuable data could fall into the wrong

1 `https://visjs.org/` [Accessed: May 31, 2023]

2 `https://docs.open-metadata.org/v1.3.x/how-to-guides/data-insights/cost-analysis` [Accessed: May 13, 2024]

3 `https://datahubproject.io/docs/generated/lineage/lineage-feature-guide/` [Accessed: May 13, 2024]

Table 6.7: DP7: Principle of Data Assessment

Design principle title	**DP7: Principle of Data Assessment**
Aim, implementer, and user	To allow the data catalog users (user) to make more informed decisions and to increase confidence (aim) in the data.
Mechanism	Provide data-related assessments, e.g., for quality, risks, integration capability, and others.
Rationale	An assessment of the data is necessary so that users of the data catalog can gain insight into the quality of the data, whether it poses a risk, complies with data governance rules, or even whether integration is easily possible. Gathering this information in one place gives users a better overview and saves time.
Extracted from	R1.1, R1.3, R1.4, R2.2, R4.1, R4.2, R8.1
Supported by	[Att15; Ayf21; Cho22; Dor21; Gon07; Has20; Ker15; Kir19a; Klí19; Kub18; Kub16; Kuč13; Lab20a; Neu18b; Neu16; Nog21; Par09; Rei22; Šli21; Zui16]
Design Features	• DF7.1: Quality Metrics • DF7.2: Review System • DF7.3: Risk Management Capabilities

hands is often considered [Moh20; Sha16a; Wan17b; Waw06]. It is rarely considered that data itself can pose a risk [Teb18, Paper II.], for example, if they suddenly become unavailable for one's project or their use in machine learning training leads to dangerous models, e.g., in the automotive or medical fields.

"From a data catalog functionality perspective, data assessment and functions can support organizations in maintaining data quality" [Lab20a, p. 208]. By offering or supporting data assessments, a data catalog can help improve data of poorer quality. Likewise, this leads to the selection of better data for one's own projects. For this, it is vital that the quality of the content is measured, and thus one can also make overall statements about the overall quality of the data catalog [Ayf21; Cho22; Gon07; Kub18; Kub16; Kuč13; Neu18b]. Here, this refers, on the one hand, to the quality of the data itself [Dor21; Has20; Zui16]. This has even been proposed as a DP in the context of ODPs by Zuiderwijk et al. [Zui16] (see Section 3.2). On the other hand, the quality, but also quantity, of metadata plays a major role [Att15; Ker15], which is why many works have addressed metadata quality issues [Kir19a; Klí19; Kub18; Neu16; Nog21; Par09; Rei22; Šli21].

6.7.1 DF7.1: Quality Metrics

To implement DP7, provide quality metrics for the data inventoried and metadata stored in the data catalog.

Metrics are one way of providing information about the quality of data. Metrics can, for example, be determined directly from the data using clearly defined algorithms, thus providing an objective perspective on data quality. Many dimensions play an important role in the question of data quality. These include, for example, accuracy, relevance, interpretability, or accessibility [Wan96]. A data catalog should provide as many quality metrics as possible. Users or other systems can use these to decide for or against data.

The quality of the data and the quality of the stored metadata also play important roles in data catalogs. The metadata is what users of the data catalog see directly and on which decisions are made. Therefore, providing users with a tool to assess the metadata's reliability makes sense. As with the actual data before, quality metrics represent such a tool.

Implementation Recommendations

For both data and metadata quality metrics, it makes sense to have them collected automatically if possible (see DF 1.2). This statement is supported by the work of Shanmugam and Seshadri, who see automated collection of data quality metrics as a possible feature of a data catalog: "A data architect could also design catalogue in such a way that it is able to [...] process [...] data from the source [...], process them against a set of quality rules [...] and do a continuous evaluation of results which can then be used to calculate or adjust ratings." [Sha16c, p. 137]. The metrics should be customizable and extensible by the users, or at least a privileged user group (see DF 2.2). Suppose the metadata model used in the data catalog is based on DCAT (see also DF 3.1). In that case, it is a good idea to consider the method of Nogueras-Iso et al. for evaluating metadata quality [Nog21]. Further guidance on the implementation of metadata quality metrics in data catalogs, particularly on a possible architectural design, is provided by the work of Kirstein et al. [Kir19b]. Overall, reference should be made again to DF 6.2 where a dashboard is recommended for getting an overview. Such a dashboard can also be helpful here, for example, to view quality metrics from a holistic perspective that spans the entire data catalog.

Application Examples

In DIVA 1.0, a solution was developed as part of DIVA Iteration 1 that allows users to compare alternative datasets based on quality metrics. The quality metrics can be maintained using the web interface (see Figure A.6) (R1.1). In the detailed view of a dataset, the values can be compared across different datasets in a radar chart (see Figure A.9). Thus, when multiple alternative datasets are available, users can choose a specific dataset based on the quality scores. Likewise, this iteration provides a management overview in the form of a dashboard that can give users an overall impression of the data catalog (R1.3)

In DIVA Iteration 4, the analyses that can be performed on tabular data have been extended (R4.1, R4.2). Simple quality metrics were added. For example, metrics to determine whether different data types were used per column or if null values exist.

One example that provides quality metrics for metadata is the EU data portal. Since its launch in February 2016, the portal has acted as a central point of contact for open data in the EU [16]. Here, metadata quality is surveyed using the so-called Metadata Quality Assurance Service. The quality metric consists of the availability of metadata and an evaluation of their content [Kir19b].

6.7.2 DF7.2: Review System

To implement DP7, provide a review system through which users can share their experiences with the inventoried data.

Aspects of data quality cannot always be expressed in number-based metrics. For example, data may have an ethical bias that is not identifiable in the metrics. However, this is something that data users may notice. They need a way to communicate their experience in this regard. Review systems in which free text is allowed are suitable.

Implementation Recommendations

The implementation and scope of functions should be based on existing solutions. On the one hand, this can be implemented in data catalogs. On the other hand, one can also orient to typical online stores. There should be the possibility to enter a free text and perform an evaluation based on points. Further features are to be designed depending on the extent of the data catalog. Attaching additional material, such as files, images, or videos, to the rating may be useful. The system must also be protected against misuse. For example, only users who have worked with the data should be able to submit a rating. If this is not technically feasible, ratings can also undergo a review process and be approved by the person responsible for the data. Also, users could flag reviews as good or bad through appropriate mechanisms. It should always be transparent to users viewing the reviews that the ratings are subjective.

Application Examples

An example of a review system in a data catalog can be found in OpenMetadata. There, a simple upvote and downvote system has been implemented. DIVA also provides such a feature, which has already been implemented as part of DIVA Iteration 1 (R1.4). Figure A.21 gives an impression of the review system in DIVA 4.x.

6.7.3 DF7.3: Risk Management Capabilities

To implement DP7, provide risk metrics so that users of the data catalog can make informed decisions based on them and take countermeasures when risks are high.

The use of data can lead to significant business risks. For example, if company processes or products are heavily dependent on the availability of certain data and regular updates. For example, if the data is generated in-house via sensor technology, it can be risky if only a single sensor generates it and fails. Suppose the data is purchased through an external provider, and it is not otherwise available in this form through other providers. In that case, this can also be a high risk if the provider suddenly stops offering it. Information

should be provided about potential risks arising from using the data. Risk metrics are an essential tool that can help to identify possible risks at an early stage. It is a good idea to store these in the data catalog or calculate them directly there. A data catalog has a complete overview of the available data and can thus calculate risk metrics with the help of the stored metadata and the data network.

Implementation Recommendations

The recommendations mentioned in implementing the quality metrics in DF 7.1 also apply. Likewise, refer to the recommendations in DF 2.2 about the flexibility of metrics creation.

Application Examples

As part of DIVA Iteration 2, DIVA was extended to include the ability to have metrics generated by users (R2.1). This was used as the basis in [Teb18, Paper II.] to allow risk metrics to be generated by users of a data catalog (R2.2). Figure A.8 shows how the interface for the creation looks like in DIVA 1.2. Figure A.9 shows how metrics were presented in the detail view of a dataset in DIVA 1.2.

End-of-data life cycle information represents another possible input for a risk metric. In Iteration 8, DIVA was extended to include the ability to model the end of the data life cycle for inventoried data (R8.1). If the end of the data life cycle is not considered, outdated, broken, or non-compliant data can be accidentally used. This represents a risk, so the end of the data life cycle should also be included in the risk assessment.

6.8 A Design Theory for Data Catalogs

We will now present our design theory for data catalogs. Concerning the structure of the design theory, we orient at the work of Jones and Gregor [Jon07]. Below, we describe the components and how we populate them.

Purpose and Scope

The class of IT artifacts for which we propose a design can be described by the term *data catalog*. We have developed the definition of a data catalog and thus the scope of the design theory in Section 2.2. *Data catalogs are metadata management tools that inventory and curate data assets by storing relevant descriptive, administrative, and structural metadata about the data and provide sophisticated functions to help those working with data match the available data demand with the existing data supply.* While the design theory can be applied to all types of data catalogs (see Section 2.2.3), it cannot be applied to data management tools that manage data directly rather than via metadata. The design theory aims specifically at practitioners who want to implement successful data catalogs. In addition, researchers will be supported in exploring further artifacts relevant to the data catalog based on the design theory.

Constructs

We build on the following constructs that are relevant to our design theory: Metadata (see Section 2.1), Data Catalogs (see Section 2.2), Automation (see Section 6.1), Flexibility (see Section 6.2), Interoperability (see Section 6.3), Context (see Section 6.4), Data Life Cycle Management (see Section 6.5), Visualization (see Section 6.6), Data Assessment

(see Section 6.7), (Meta-)Data Management, FAIR Principles, Data Governance, Data Democratization (see Chapter 1 and Chapter 3).

Principles of Form and Function

We extracted seven DPs from a data catalog developed over six years in collaboration with practitioners. The DPs are: DP1 Principle of Automation, DP2 Principle of Flexibility, DP3 Principle of Interoperability, DP4 Principle of Context, DP5 Principle of Data Life Cycle Management, DP6 Principle of Visualization, DP7 Principle of Data Assessment.

Artifact Mutability

We focus on describing the inevitable changes of the artifact over time, which are taken into account by our design.

AM1 *Mutability of Typology.* Data catalogs are used in various scenarios. They are utilized in businesses, the public sector, and other settings, such as data spaces. Data catalogs are used differently in each of these scenarios. Users, for example, have varying expectations for the metadata and analytics offered. Data catalogs must be able to accommodate these various requirements. It can also be assumed that new types of data catalogs will emerge in the future, which we do not currently know. Data catalogs must be able to adapt to these new types. Due to the strong abstraction of the DPs, we assume that data catalogs that implement our design can also be used in future scenarios.

AM2 *Mutability of Data Landscape.* Data landscapes in different contexts are highly heterogeneous. Data catalogs must be able to embed themselves in these data landscapes and interact with them. They must be able to inventory the data and enable (meta-)data management. They must support the various standards and formats used in the data landscapes to do this. We anticipate that new technologies, such as data formats or interfaces, will continue to be developed. Data catalogs must be able to adapt to these changes. Data catalogs that implement our design, especially DP2 Principle of Flexibility and DP3 Principle of Interoperability, should be able to adapt to these changes.

Testable Propositions

The Testable Propositions (TPs) that can be used to test if our design is successful, are the following:

TP1 *A data catalog that implements our design supports faster data discovery.* We have already mentioned in the motivation for this work that data scientists and knowledge workers often have to spend a long time searching for the data they need. In particular, the mass of data, its heterogeneity, and the formation of data silos complicate the search for data. Data catalogs that implement our design (especially DP1, DP3, and DP4) can create an inventory of the data that is as complete as possible. This implies that sufficient metadata and contextual knowledge exist to describe the data, thus improving data discovery and fostering reusability.

TP2 *A data catalog that implements our design supports the FAIR principles.* Making data findable, accessible, interoperable, and reusable is a central goal in many data management initiatives. Data catalogs that implement our design (especially DP1, DP3, DP4, DP5, and DP5) are able to support the FAIR principles. Data, in the form of metadata, is indexed in

the data catalog, which makes it findable. To make data accessible, technical information regarding the permission processes, access protocols, and accountable parties, for example, can be provided. Data can be made interoperable by offering details about data formats, schemas, or models, for example. Data is reusable by, e.g., providing the necessary context information like licenses, data lineage, or usage policies.

TP3 *A data catalog that implements our design supports data democratization.* Data Democratization is an essential goal in many data-driven organizations. The goal is to make data available to all employees and enable them to work with it. Both technical and non-technical users should be able to understand the data better and select appropriate data. Data catalogs that follow our design (especially DP4, DP5, DP6, and DP7) will provide the necessary mechanisms that makes understanding and selecting data easier.

TP4 *A data catalog that implements our design supports data governance.* Data Governance is an important factor in data management, as it supports aspects like data quality, data integrity, and data privacy. The goal is to have a consistent and reliable data management process that specifies responsibilities and the rules and regulations it must follow. Data catalogs that follow our design (especially DP4, DP5, and DP7) will support data governance. For example, metrics determine data quality and user ratings and problems can be reported to the responsible persons. Responsible persons can also manage various phases of the data lifecycle in the data catalog. Regulatory guidelines can be stored in the data catalog as context information to support compliance.

TP5 *A data catalog that implements our design can be easily adapted to the needs of the users and the environment.* In detail, a data catalog's necessary functionalities and features differ depending on the context. This applies, for example, to the data-holding systems to be connected, which visualizations are required, or which metrics are to be collected. Data catalogs that adopt our design (especially DP2) are flexible to the needs of their users and the environment in which it is deployed. They can, for example, adapt to specific access regulations, company-relevant metrics, or context information.

Justificatory Knowledge

In the context of this work, justificatory knowledge is constructed through informal knowledge [Gre13a]. We use the accumulated field knowledge we gathered during the six years of development of DIVA. This knowledge includes the expertise of the numerous practitioners who have supported us throughout the development of DIVA. They were essential in refining and improving the design step by step. Likewise, they played a significant role in evaluating our prescriptive design knowledge (see Chapter 7) and thus formed the basis for our justification.

Principles of Implementation

We developed 18 DFs that assist in instantiating the design. They concretize the implementation of the DPs and support practitioners in implementing data catalogs. The DFs are: DF1.1 Automated Inventory, DF1.2 Automated Metadata Gathering, DF2.1 Extensible Metadata Model, DF2.2 Creation and Customization of Metrics, DF2.3 Customizable Policy System, DF2.4 Configurable Workflow Engine, DF3.1 Standardized Metadata Models, DF3.2 Standardized API Documentations, DF3.3 Open Source, DF4.1 Metadata With

Contextual Information, DF4.2 Data Networks, DF5.1 Data Usage Policies, DF5.2 End-of-Life Data Management, DF6.1 Metadata Visualization, DF6.2 Data Network Visualization, DF7.1 Quality Metrics, DF7.2 Review System, DF7.3 Risk Management Capabilities.

Expository Instantiation

Since we have extracted the DPs and DFs from DIVA, the software can be regarded as expository instantiation. DIVA can be used by practitioners to get deeper insights into our design theory, thus supporting its realization. We discussed the DIVA development in Chapter 5. In addition to DIVA, the DAC and the software for generating Destroy Claim Agents can also be regarded as expository instantiation. These provide further insights into how DF1.1 Automated Inventory, DF1.2 Automated Metadata Gathering, and DF5.2 End-of-Life Data Management can be implemented in practice.

CHAPTER 7

Evaluation

"Evaluation, in design science research, is usually regarded as a validation or proof process where a design theory is shown empirically to stand in terms of how well a resulting artifact performs or to what degree it works as intended" [Lee11, p. 9]. Evaluating artifacts is crucial to ensure scientific rigor in DSR. It allows for determining whether the set goals have been achieved and provides information on what can be improved in future iterations of the artifact. It should also be examined whether the results are novel and innovative. "[...] [I]t is important to keep in mind that what is new and innovative may differ in the case of researchers and practitioners." [Iiv18, p. 8]. Only through evaluation can it be ensured that the results are relevant for science and practice [Iiv21]. It is therefore inevitable that the "[...] utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods" [Hev04, p. 12]. Literature has established several evaluation methods in the context of DSR. These include, for example, case studies, field studies, experiments, analytics, or simulations [Hev04].

To evaluate our design theory, we focus on evaluating the DPs, because they are the most important component of a design theory [Hei14; Jon07]. We used different methods to obtain a complete and comprehensive assessment of the DPs. On the one hand, we check the reusability of the DPs based on the work of Iivari et al. [Iiv21]. On the other hand, we check the ontological expressiveness of DPs according to the method of Recker et al. [Rec11] and Janiesch et al. [Jan20]. This chapter will first describe our evaluation design and explain the two methods mentioned. We will then discuss the data collection and present the results. Finally, we discuss the results and conclude the evaluation.

7.1 Evaluation Design

According to Venable et al., two main dimensions characterize evaluation in DSR [Ven16]. First, they distinguish between artificial and naturalistic evaluation. Artificial means that the evaluation occurs within a strictly predefined and controllable setting. In most cases, only a tiny part of the reality is modeled. This type of evaluation is easy to plan and execute. Accordingly, others can more easily understand, repeat, and confirm or refute the evaluation. Naturalistic means an evaluation in a real environment, such as a company. Since naturalistic evaluations can be time- and resource-intensive, they may be omitted, and no evaluation is performed. However, "[...] the neglect of reusability evaluation increases the risk of publishing DPs that are not found applicable by practitioners and not useful in practice" [Iiv21, p. 286]. Second, a distinction is made between formative and summative evaluation. In principle, a distinction is made between the purpose for which the evaluation is carried out. Formative evaluation usually takes place during the DSR iteration cycles

and aims to find out which aspects of the artifact can be improved in future cycles. The summative evaluation is often found at the end of the DSR project and tries to provide a final assessment of the artifact. We have decided to perform a light-weight formative evaluation in an artificial setting. This allows us to gain an initial insight into whether our DPs, and therefore our design theory, can be used meaningfully in practice.

Applied Methods

As the first evaluation strategy, we applied the light evaluation framework for DPs according to Iivari et al. [Iiv18; Iiv21]. The evaluation was done artificially without the increased effort of a necessary instantiation. The evaluation is based on five criteria that need to be evaluated. A positive evaluation of the five criteria indicates good reusability, usefulness, and relevancy of the DPs [Iiv21]. The criteria are briefly described below:

(1) Accessibility refers to how understandable the DPs are formulated for the target group. For example, whether they are well-structured, use terminology appropriate for the target group, and whether the target group can understand the DP. *(2) Importance* refers to the relevance of the DPs and the resulting class of artifacts they produce. In this context, different practitioners may assess the importance differently. Thus, DPs that were shaped and validated by practitioners at development time may be identified as unimportant by practitioners in the evaluation phase. *(3) Novelty and Insightfulness* describes that the DPs must provide new knowledge for the practitioners and not only reflect what is already known and applied daily. This criterion should be considered in particular because while novelty is often the focus of research, it is often neglected for practice in the context of DSR. *(4) Actability and Guidance* states that practitioners can implement the DPs in the real world. Particular attention must be paid to ensuring that the DPs are realistic. *(5) Effectiveness* deals with the impact of the DPs and the artifact instantiation on the practitioners and their environment. Here, for example, questions arise about increased performance or productivity. However, measuring these is difficult and requires instantiating the artifact with the help of the DPs in the real world for concrete statements.

The second evaluation strategy we applied was the ontological expressiveness assessment for DPs by Janiesch et al. [Jan20]. They constructed an ontological expressiveness assessment for DPs based on the work of Recker et al. [Rec11]. In their work, Recker et al. argue that the grammar of a modeling language determines the outcome of a modeling process. Similarly, Janiesch et al. apply the same logic to DPs as they determine the outcome of a design process. Therefore, they reused the evaluation of modeling grammar of Recker et al. [Rec11]. The basic idea is that there is a one-to-one correspondence between real-world phenomena important to the design of an artifact and corresponding DPs. The absence of deficiencies strongly indicates that sound DPs have been identified. In sum, there are four types of deficiencies that need to be assessed. Figure 7.1 gives a visual impression of the deficiencies. In the following, we want to explain the deficiencies briefly.

(1) Principle Deficit means that not all DPs could be identified during the generation, which are necessary to describe existing and important phenomena. At this point, the target group lacks prescriptive knowledge to implement a design that works in practice. Recker et al. identified that "deficit motivated [...] users to employ additional means

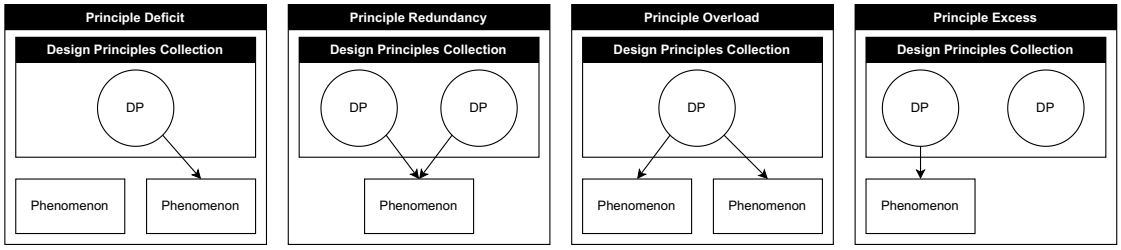


Figure 7.1: Assessment criteria of the DPs expressiveness [Jan20]

to help articulate the real-world phenomena they felt could not be expressed" [Rec11, p. 59]. *(2) Principle Redundancy* describes that more than one DP addresses the same phenomenon. Therefore, the target group does not know which DP primarily achieves the effect and whether these DPs should be combined. *(3) Principle Overload* describes that a DP covers too many phenomena. One can already speak of overload if the DP describes more than one phenomenon. One can assume that "[...] overload undermined users' ability to understand the information contained [...]" [Rec11, p. 59]. *(4) Principle Excess* means that more DPs have been elaborated than would be needed to implement a relevant design. Thus, there are more principles than there are phenomenons. Recker et al. have identified that "[...] excess [...] resulted in users misunderstanding [...]" [Rec11, p. 59].

Data Collection

To conduct our evaluation, we selected individuals from the target group of our DPs. We were able to recruit eight participants for the evaluation. These individuals collectively cover all types of data catalogs we investigate in this work (see Section 2.2.3). The participants were sent information material in advance (see Appendix B), which, on the one hand, briefly explains the purpose of the research project and, on the other hand, contains the DPs and associated DFs in tabular form. Table 7.1 lists the persons and briefly characterizes them.

Table 7.1: Characterization of the participants

#	**Role**	**Experience**
1	Software Engineer, Research	Six years of professional software engineering experience in industry projects. Responsible for specifying and implementing an industry client's **ecosystem data marketplace**.
2	Lead Software Engineer, Automotive	Seven years of professional software engineering experience in industry projects. Developing and contributing to establishing company-wide data pipelines and **enterprise data catalogs**.

Continued on next page

Table 7.1 – continued from previous page

#	Role	Experience
3	Software Engineer, Cloud Provider	Seven years of experience in professional development of software solutions for industry and research partners. Currently responsible for selecting and enacting an **enterprise data catalog** for their cloud product portfolio.
4	Software Engineer, Retail Industry	Nine years of professional software engineering and consulting experience. Experience and involvement in data management and **enterprise data catalog** topics.
5	Data Catalog Consultant, Research	Three years of experience in data catalog consulting with partners from the industry (**enterprise data marketplace**, **enterprise data management platform**, **ecosystem data marketplace**). Responsible for developing the specifications for an industry client's ecosystem data marketplace. Involved in research for **data space data catalogs**.
6	Librarian, University Library	Seven years of experience working with metadata management, focusing on literature. Participates in developing improvements to **context-specific data catalog** software and related systems, focusing on recommender systems.
7	Software Engineer, Cloud and Retail Supply Chains	Thirty years of open-source experience. Experience in cataloging physical assets. Involved in data space topics for the last two years and in the selection and development of **data space data catalogs**.
8	Data Engineer, Cities and Municipalities	Four years of experience in consulting and requirements engineering with cities and municipalities in the context of data platforms and **ODPs**.

The first part of the evaluation was conducted using a questionnaire sent to the participants (see Appendix C). The questionnaire was implemented using LimeSurvey[1]. We took the questions from the work of Iivari et al. and adapted them to our context [Iiv18; Iiv21]. As suggested by Iivari et al., we adjusted the questions in the *Effectiveness* group to fit our situation. For example, we changed the questions of type "Compared to my current situation, I believe that [Type X system] would improve [...]" to "I believe that the design principles can help design data catalogs that improve [...]". The questionnaire contains 22 questions in the form of a 5-point Likert scale. The answer options are *strongly disagree*, *disagree*, *neither agree nor disagree*, *agree*, and *strongly agree*. The questionnaire aims to

1 `https://www.limesurvey.org/` [Accessed: August 19, 2023]

assess the totality of the DPs for data catalogs and check for reusability, usefulness, and relevancy. Therefore, the DPs are considered a single unit in the questionnaire, and there are no questions on individual DPs or DFs.

After the questionnaire, the participants were invited to a one-on-one expert discussion. The appointments were carried out remotely and lasted about an hour each. For the execution, we once again followed the work of Iivari et al. Here, the DPs are considered individually, and the five criteria are checked to identify possible outliers. Subsequently, according to Janiesch et al., the deficiencies are addressed by discussing whether Principle Deficit, Principle Redundancy, Principle Overload, or Principle Excess is present. Due to time constraints, not all aspects can always be addressed in conversations. The participants dictated the focus. We guided the conversation so that as many aspects as possible were considered. The conversations were recorded, and the most relevant aspects are summarized in Appendix D.

7.2 Individual Evaluation of the Design Principles

This section addresses the evaluation of the individual DPs. For this purpose, we studied the expert discussions (see Appendix D). We checked them for occurrences of the five criteria according to Iivari et al. [Iiv21]. If a criterion was mentioned positively, it was marked with a "+". Negative expressions were marked accordingly with a "−". If a criterion was not mentioned, or there were both positive and negative mentions, the criterion was marked using a "○". The result of this process can also be seen in Appendix D. For a better overview, the results regarding the five criteria according to Iivari et al. are visualized as a stacked bar chart using the matplotlib[1] library. A separate visualization was generated for each DPs. The following discusses the results of the individual DPs.

7.2.1 Principle of Automation

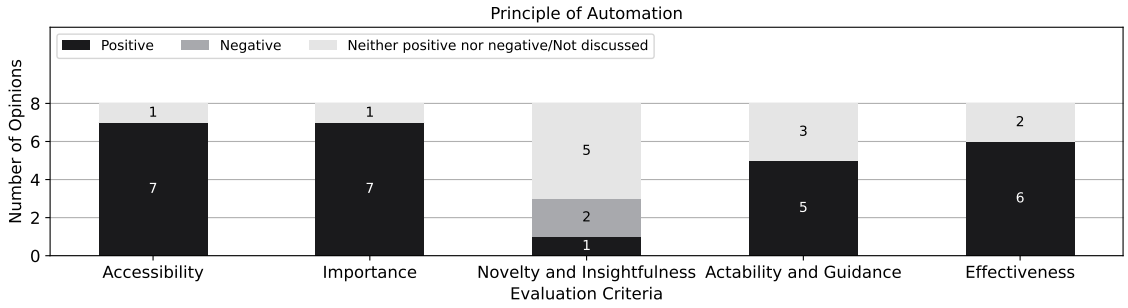


Figure 7.2: DP1 expert discussion evaluation results

Figure 7.2 gives an overview of the criteria according to Iivari et al. regarding the Principle of Automation. As can be seen, the participants see the DP as understandable. The importance is also rated as high by the participants. Automation is an important factor in saving time and resources (E1, E3, E5, E6, E8). E1 specifically describes that automation plays an important role for operators of data marketplaces. In particular,

1 `https://matplotlib.org/` [Accessed: September 30, 2023]

finding a way to adjust prices for data according to the market situation automatically was mentioned here. E2 sees automation as a core issue for enterprise data catalogs. Many things must be automated for the data catalog to work (E2, E3). Automation should be seen as an ongoing task that is always being pushed forward (E2).

At this point, making precise statements about novelty and insightfulness is impossible since no tendency was discernible in five conversations. However, according to E2, much is already done in this area. E7 assumes that there should always be a high degree of automation. E8 does not see automation as a new topic. Nevertheless, there are new insights in the context of data catalogs, especially in interaction with the other DPs. While the topic itself is not new, it is important in the context of data catalogs to point out that efforts should always be made to automate processes. E3 appropriately points out that it would not work without automation, for example, with cloud providers, due to the mass of existing data. Even if data catalogs can be handled manually on a small scale, automation often makes sense. Especially with data catalogs, this DP can stimulate thinking about which aspects can be automated.

The participants see the actability and guidance of the DP as generally good. However, it is correctly pointed out that it is a continuous (E2) and costly (E8) process. Also, not everything can be trivially automated (E1, E4, E7), so automation is neglected in the worst case (E7). The participants also support the effectiveness of the DP. It would allow users to spend their workforce on other things (E1, E6), as they would save time (E3, E7).

In summary, it can be said that the reusability of the Principle of Automation is given due to the mostly positive assessments of the criteria. Although there are no novel insights for the target group at a high level, this does provide another impetus to address the topic in the context of data catalogs.

7.2.2 Principle of Flexibility

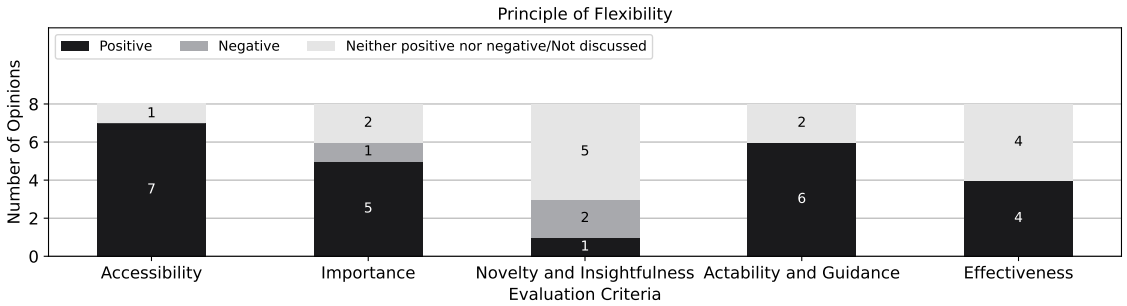


Figure 7.3: DP2 expert discussion evaluation results

Figure 7.3 gives an overview of the criteria according to Iivari et al. regarding the Principle of Flexibility. As with the Principle of Automation, the Principle of Flexibility is also considered to be understandable and comprehensible by the participants. There was no negative response, and 7 out of 8 participants expressed a favorable opinion. The importance is supported by most participants (E1, E2, E3, E5, E8). Flexibilization is important so that one does not have to make one's own adjustments to the code (E1, E5). For example, new metadata can be stored, and data sources can be connected without

much effort (E1, E3, E8). The ability to turn off features on demand is also part of this (E5). Enabling flexibility makes sense due to the complexity of data catalogs and the various use cases (E1, E2, E3). However, participants also point out that flexibility is not implemented or considered important in case of doubt. This can be the case if there is no added value (E1, E6) or if one has expertise in customizing software code (E3).

No definitive answer can be given about the novelty and insightfulness of the DP since most participants did not comment positively or negatively on this. However, it can be said that the flexibility of software systems is nothing new in itself (E1, E7), but it is well justified in the context of data catalogs (E8). The participants estimate the actability and guidance of the DP as realistic and reasonable (E2, E3, E4, E6, E7). However, several participants pointed out that implementation can be challenging (E4), expensive (E3), and time-consuming (E1). In terms of effectiveness, there is no absolute positive majority. Nevertheless, at least half of the participants expressed positive opinions in the discussions. For example, implementing the DP can save time and resources (E5, E7) because the user can extend the data model by himself (E1). The various requirements that companies or governmental institutions have, for example, would also be easier to meet (E8).

In summary, the reusability of the DP is given. However, it also becomes clear that it must be carefully considered when to implement the principle and when it is not worthwhile. It cannot be blindly applied to everything since the feasibility and the added value are sometimes questionable. Our DFs, including reasonable use cases, can aid decision-making.

7.2.3 Principle of Interoperability

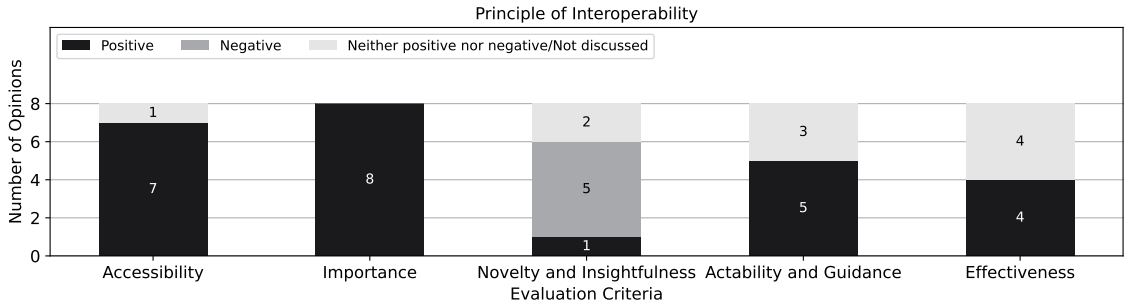


Figure 7.4: DP3 expert discussion evaluation results

Figure 7.4 gives an overview of the criteria according to Iivari et al. regarding the Principle of Interoperability. The Principle of Interoperability is assessed as understandable by the participants. All participants mentioned the importance of the principle positively. For example, it is considered vital for data catalogs to exchange metadata with each other (E7, E8) or to be merged in the case of a company consolidation (E5). Also, the DP is seen as important to meaningfully implementing the Principle of Automation. These two principles complement each other well (E2). Interoperability in the form of FOSS is also seen as important to adapt the code of the data catalog to one's needs (E3).

The Principle of Interoperability has received the most negative mentions regarding novelty. In this case, they even make up the absolute majority. The DP is not considered novel by the participants, nor does it provide much in the way of new insights that affect

previous work. Interoperability needs to be considered very often when implementing software and is already being considered (E1, E3, E8). It is also nothing new in the area of data catalogs (E4), as standards such as DCAT are also already used in practice (E8).

The participants consider the DP realistically implementable (E1, E2, E4, E7, E8). However, it can be difficult if, for example, many systems have to communicate with each other (E1) or new systems must be connected for the first time (E8). In practice, especially DCAT is used as a standardized data model (E8). The effectiveness of the DP is confirmed by half of the participants. On the one hand, possible time savings are mentioned (E7, E8). E6 even states that two colleagues could devote themselves full-time to other important things if interoperability becomes more common. On the other hand, it is mentioned that interoperability makes collaboration meaningful in the first place (E1) and that nothing would be feasible in large companies without interoperability due to the number of systems (E5).

In summary, the reusability of the Principle of Interoperability is given. However, it is also clear that no new knowledge is available for the target group on this abstract level. Nevertheless, all participants commented positively on the importance of this DP and that it should be kept. Even if it could provide little to no new knowledge at this level, it still represents an important aspect of data catalogs. Also, in the interaction with the Principle of Automation, it becomes clear how much one must pay attention to interoperability. A lack of interoperability makes automation considerably more difficult.

7.2.4 Principle of Context

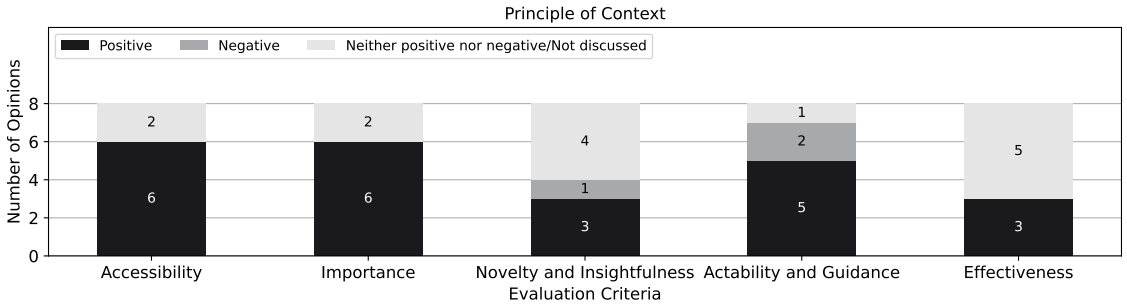


Figure 7.5: DP4 expert discussion evaluation results

Figure 7.5 gives an overview of the criteria according to Iivari et al. regarding the Principle of Context. As can be seen, the DP is formulated in a way that the participants understand. The DP is also seen as important. Context information can be crucial for deciding whether a data purchase takes place in ecosystem data marketplaces (E1). There is an increasing trend toward introducing data networks in context-specific data catalogs such as those used by libraries, for example (E6). An increasing number of providers are introducing these and offering improved search capabilities. The provision of contextual information is seen as an essential component for a data catalog to be helpful (E2, E4, E5).

Regarding novelty and insightfulness, no absolute majority exists. One participant sees context as something that already occurs in many domain-specific models and is known (E3). Another participant has gained new insights through the DP, as he otherwise takes

a rather technical perspective and is rarely confronted with the topic of context (E7). Similarly, the creation of a data network (DF 4.2) is described as something truly new emerging, both in context-specific data catalogs and ODPs (E6, E8).

The question of actability and guidance regarding the DP is answered positively by an absolute majority. Two participants were rather negative about the feasibility. For example, it is considered challenging to monitor the data lineage of all inventoried data (E1). The DFs could be of more help in answering the question of what context should be offered, for example, in ecosystem data marketplaces. However, it is recognized that answering this question in the context of a DP is beyond the scope (E1). Even in the context of ODPs, the context issue is not always trivial. For example, there is a concern that if automatisms are missing, users will not fill in the context data by hand, and it cannot be enforced on them (E8). E3 is optimistic about this but notes that one must apply the Principle of Flexibility to add new context information at runtime. Like E1 and E8, E5 sees automation as an important support to implement the DP.

Most participants did not comment on the effectiveness of the DP. Therefore, we can only assume that implementing the DP positively affects successful data catalogs. Thus, it is said that the data can be used better due to the context (E3), or the search function, the main feature of a data catalog, can function better (E6).

In summary, the participants predominantly see the DPs reusability. In detail, questions have arisen that the DPs and the DFs do not answer. However, this is primarily because a conclusive and comprehensive answer would go beyond the scope. Thus, implementation still requires active input from the target group to identify useful contextual information. How this context is obtained also needs to be identified during implementation.

7.2.5 Principle of Life Cycle Management

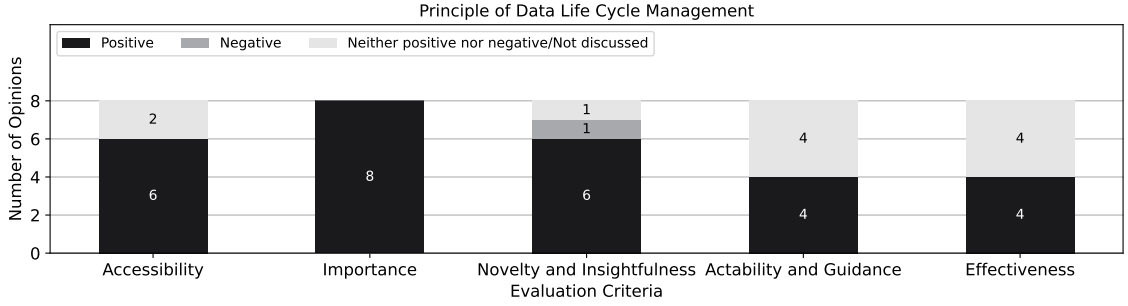


Figure 7.6: DP5 expert discussion evaluation results

Figure 7.6 gives an overview of the criteria according to Iivari et al. regarding the Principle of Data Life Cycle Management. The participants understand the DP, and no questions have arisen regarding the subject matter. All participants commented positively on the importance of the DP. Thus, data life cycle management can be relevant and helpful in many situations and generally plays an important role (E3). Making it explicit at a central point (E3, E5) rather than just implicit (E4) is useful. In ecosystem data marketplaces, it can help to avoid offering expired data further by mistake (E1). Library catalogs can be used to manage when licenses for articles and other works expire and must

be deleted from the servers (E6). For ODPs, license management is also important to know how to use the data (E8). The lack of data life cycle management can also lead to an unnecessary, useless data catalog (E2). One should also consider managing the data and metadata lifecycles (E8).

With six positive comments, the novelty and insightfulness of this DP stand out. Participants gained new insights into how they should implement data catalogs. Only one participant generally considered the management of assets familiar and saw no novelty (E3). The positive comments were mainly about the fact that it is a novel and valuable idea to centralize everything in a data catalog (E5, E6, E7). In particular, it is also a novel and relevant idea to integrate data life cycle management in data marketplaces (E1) and ODPs (E8).

Half of the participants expressed a positive opinion on actability and guidance, while the other half did not express any opinion. For the participants, it is clear how they would proceed, even if there could still be questions about the implementation details (E4). For example, some work on data life cycle management already exists and can be transferred to the context of data catalogs (E6). Data life cycle management must also be practiced outside the data catalog to ensure proper operation (E3).

As with the previous criterion, half of the participants expressed a positive opinion on effectiveness, while the other half did not have any opinion. Implementing the DP would save users time (E5, E8). Also, the lack of life cycle information would render the data catalog useless, negatively impacting user performance and productivity (E2). However, it is also essential to consider that the data owner thus potentially has more work to maintain the data life cycle in the data catalog (E3).

In summary, the reusability of this DP is given. The participants perceived it as very useful. In contrast to the other DPs, what is striking here is the high level of agreement regarding the novelty. Combined with the high importance, it becomes clear that this DP is essential for data catalogs.

7.2.6 Principle of Visualization

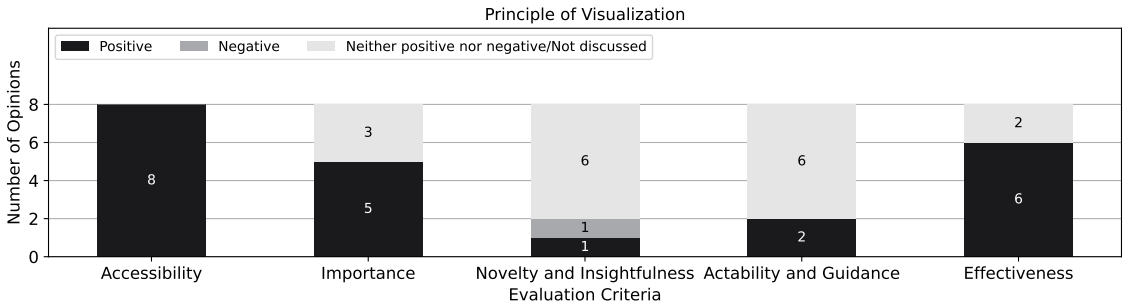


Figure 7.7: DP6 expert discussion evaluation results

Figure 7.7 gives an overview of the criteria according to Iivari et al. regarding the Principle of Visualization. All participants confirmed the clarity of the DP. However, E4 suggests a renaming. E4 sees the focus primarily on creating an understanding of data and not on visualization. For example, the Principle of Comprehension or Principle of

Understanding were suggested as names. However, since the principle only aims at visual support and does not address, for example, traceability or explanations in text form, a renaming would imply more scope than is intended.

The absolute majority of the participants expressed themselves positively regarding the importance of the DP. Visualizations are important for enterprise data catalogs (E5), data marketplaces (E1), context-specific data catalogs (E6), data space data catalogs (E7), and ODPs (E8). Visualizations can make complex relationships comprehensible (E5) and provide overviews (E6). Also, excellent and extensive visualization possibilities can be a selling point for data catalogs (E8).

We cannot report on the novelty and insightfulness of the DP here since the absolute majority of the participants did not address this criterion in the conversations. Only E3 commented that, in general, software with graphical user interfaces should always use visualizations where appropriate, and thus the principle is not new. E8 sees the DP as a novelty since it is not yet implemented in the ODP domain, and thus potential is lost.

The topic of actability and guidance was also not discussed extensively in the conversations with participants. Six participants expressed neither a positive nor a negative opinion. Two participants expressed somewhat positive views on the criterion. Both believe that visualization at a basic level is possible but that use case-related visualizations would have to be generated individually (E3, E4).

Regarding the effectiveness criterion, the absolute majority expressed a positive opinion, and nobody expressed a negative one. The participants see visualizations as a factor that simplifies the work with data catalogs (E1, E4, E8) as long as they are flexible (E3). Complex relationships, such as data lineage and linkage, can be grasped more quickly (E5, E7).

Of all the DPs, we can make the least reliable statements about the usefulness and reusability of this one. Although participants confirmed the effectiveness of the DP, little was generally reported by participants about importance, novelty, insightfulness, actability, and guidance. However, based on the consistently positive comments, the DP can be classified as reusable and relevant rather than not.

7.2.7 Principle of Data Assessment

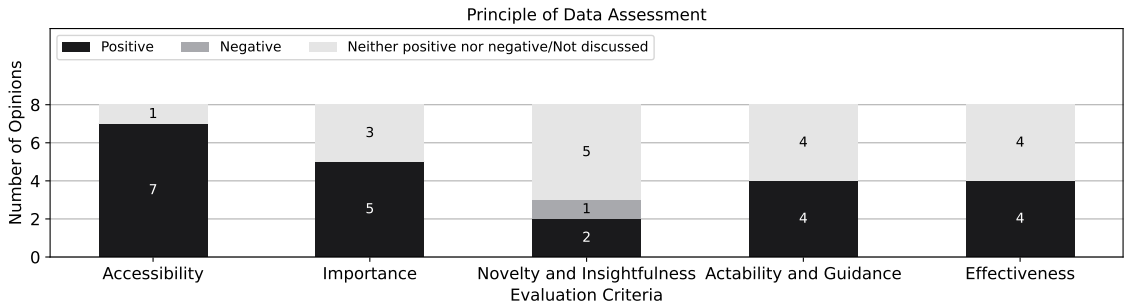


Figure 7.8: DP7 expert discussion evaluation results

Figure 7.8 gives an overview of the criteria according to Iivari et al. regarding the Principle of Data Assessment. First, the participants understood this last principle. An

absolute majority sees the DP as important. None of the participants made a negative statement. If a data catalog implements the principle, one can preserve, ensure, obtain, and measure data quality in a single place (E3). Trust in the data is also seen as important (E5). This capability is vital in ecosystem data marketplaces so potential buyers can get a sample. A possible DF could be to provide a buyer's guide that supports users in selecting the best data (E7).

No definite conclusion can be made about the novelty of the DP. Five of the participants did not take a position in the conversation. E3 sees an assessment for important assets, whether data or, for example, physical goods, as important and says that this must be done and is nothing new. Participant E6 sees the DP as new for context-specific data catalogs used by libraries. However, only experienced users would benefit (E6). Also, with ODPs, the DP is judged as new since many ODPs do not do assessments (E8).

Regarding actability and guidance, we have four neutral and four positive statements. The positive voices are all similar concerning implementation. Essentially, it is seen as possible. However, the large variety of data makes it difficult to generate suitable assessments for all situations (E1, E2, E4, E8). Limits must be established for the implementation of assessments (E4). One should see the DP as a continuous task that has to be carried on and on (E2). Also, evaluating data quality is not always trivially possible (E3) since it is also not always clear what quality means in the respective situation (E8). However, straightforward solutions, such as a review system, should be implemented (E3).

As with the previous criterion, four participants expressed neither positive nor negative opinions, and four expressed positive opinions regarding the effectiveness. It is stated that the design implementation leads to a faster selection of the appropriate data by the users of a data catalog (E1) since they immediately see whether they are correct, for example (E8). It is also argued that reliable assessments increase the probability of selecting the best data for a project (E3). However, it should be noted that potentially only experienced users benefit from this (E6).

In summary, the reusability of the Principle of Data Assessment can be seen as given. While no firm conclusions could be made about whether the DP is novel for data catalogs, the other criteria can be seen as positively expressed.

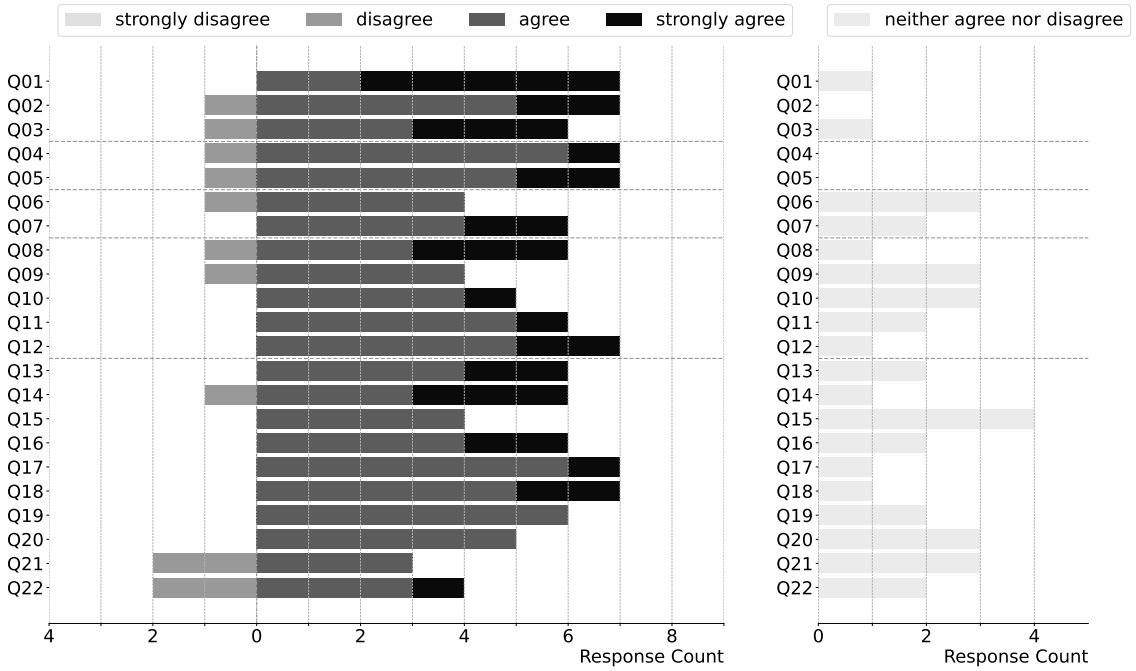
7.3 Overall Evaluation of the Design Principles

Now, we will look at the evaluation results that allow us to make a statement about the entirety of DPs. First, we evaluate the questionnaire and try to get an initial picture of the totality of the DPs. The subsequent section compares the questionnaire with the experts' statements in the discussions. After that, we elaborate on possible deficiencies identified by the participants. Finally, we present general expert feedback.

7.3.1 Evaluation of the Questionnaire

In this section, we deal with the evaluation of the questionnaire. For the visualization, we used a diverging stacked bar chart to better visually compare the positive and negative responses. The neutral responses *neither agree nor disagree* were moved outside. The chart can be viewed in Figure 7.9.

First, we will provide a global overview of the results. The answer option *disagree* was



Accessibility:
Q01 The design principles are easy for me to understand
Q02 The design principles are easy for me to comprehend
Q03 The design principles are intelligible to me

Importance
Q04 In my view data catalogs address a real problem in my professional practice
Q05 In my view data catalogs address an important - acute or foreseeable - problem in my professional practice

Novelty and insightfulness
Q06 I find that the design principles convey new ideas to me
Q07 I find the design principles insightful to my own practice

Actability and appropriate guidance
Q08 I think that the design principles can be carried out in practice
Q09 I find that the design principles provide sufficient guidance for designing data catalogs
Q10 I find that the design principles provide sufficient direction for designing data catalogs
Q11 I find that the design principles are not restrictive when designing data catalogs
Q12 I find that that the design principles provide me with sufficient design freedom when designing data catalogs

Effectiveness
Q13 I believe that the design principles can help design data catalogs in practice
Q14 I find the design principles useful for designing data catalogs in practice
Q15 I believe that the design principles can help design data catalogs that improve the performance of its users
Q16 I believe that the design principles can help design data catalogs that improve the productivity of its users
Q17 I believe that the design principles can help design data catalogs that improve the effectiveness of its users
Q18 I believe that the design principles can help design data catalogs that improve the quantity of work of its users
Q19 I believe that the design principles can help design data catalogs that improve the quality of work of its users
Q20 I believe that the design principles can help design data catalogs that improve the innovativeness of an organization/company
Q21 I believe that the design principles can help design data catalogs that improve the reputation of an organization/company
Q22 I believe that the design principles can help design data catalogs that improve the job morale of an organization/company

Figure 7.9: Results of the questionnaire

used only 12 times out of 176 possibilities. *strongly disagree* was not even chosen once by the participants. The answer option *neither agree nor disagree* was chosen more often but did not reach an absolute majority for any question. Out of the 22 questions, there are only five questions where the positive response does not reach the absolute majority. Thus, the survey gives a positive picture of the DPs. We will address the individual evaluation criteria in more detail in the following.

Q01-Q03 are questions about the **accessibility** of the DPs, i.e., whether they are understandable. Q01 is also the question with the most *strongly agree* answers. Q02 and Q03 are also very positive here, even though they each have one *disagree* as a response. In detail, it is apparent that most participants understand the DPs, but a deeper comprehension needs to be developed. This may be because the DPs are kept very abstract and thus provide less information for a deeper understanding. We can nevertheless assume that the target group understands the DPs well enough. This is an essential foundation. If the participants did not understand the DPs, all further statements on the follow-up criteria would not be nearly as relevant in significance.

Q04 and Q05 are questions about the **importance** of data catalogs in the participants' professional work. Both questions were positively answered, indicating that research on data catalogs is also relevant to practice. Interestingly, however, both questions also had one *disagree* response. One or two participants do not see data catalogs solving a real problem in their practice. This answer is interesting because we extracted the participants from a population that develops and implements data catalogs. Thus, we initially assumed that data catalogs would naturally be viewed as something that solves existing problems. However, the question explicitly targets the participants' work. A developer of a data catalog may grasp the need for it. However, a data catalog does not necessarily help that particular person do his or her job. Despite the two *disagree* answers, we can still support the importance of data catalogs in general.

Questions Q06 and Q07 are about the **novelty and insightfulness** of the DPs. Whether the DPs give the participants new knowledge for implementing data catalogs or whether they get better insights for their practical work. The positive expression is not as strong as in the previous two criteria. Many more *neither agree nor disagree* responses were given. This may be because the participants could not make a uniform statement about all DPs. A more detailed insight can be taken from the individual evaluation in Section 7.2. It has already been identified that the novelty and insightfulness of individual DPs differ significantly.

The next category, **actability and guidance**, comprises questions Q08-Q12. It can be said that the participants consider the DPs to be implementable. Q08 has received one *disagree* and one *neither agree nor disagree*. In contrast, there are three *agree* and three *strongly agree*. We believe the two non-positive responses refer to individual DPs rather than the entirety. Q09 also contains one *disagree* and even three *neither agree nor disagree* criteria. Thus, it is unclear whether our DPs provide enough implementation guidance. The question here is how to implement the DPs. We have also attached the corresponding DFs in the information material (see Appendix B). It seems that this was not always sufficient to answer the questions. Within the scope of this thesis, the DPs and DFs are elaborated in much greater detail. This level of detail would have been too much for the

information material. Q10 is more positive than Q09. Both suggest that the DPs are trending in the right direction but that participants would like more information. Q11 and Q12 are again answered very positively. Thus, participants see our DPs as flexible and can develop their data catalog with enough flexibility. This is positive because practitioners can easily incorporate their experience into their development without restricting themselves too much. This positive characteristic of Q11 and Q12 is undoubtedly also because the DPs are kept abstract. Thus, it is natural for the implementers to add details.

The **effectiveness** criterion comprises questions Q13-Q22. The purpose here is to clarify whether the DPs help to build better data catalogs than they would be without them. First, the DPs are rated as helpful (see Q13). The participants rated the usefulness of the DPs positively. *disagree* was only selected once. Questions Q15-Q19 focus on improvements that can be experienced by a user of a data catalog that implements our DPs. Q15 and Q16 represent an interesting pair. Q15 refers to user performance, and Q16 refers to user productivity. As one can see, productivity is rated more positively than performance. Performance here refers to the quality of the users' results, and productivity refers to the quantity of the users' results. The participants agree that both are increased, but especially the quantity of the results. A data catalog that implements our DPs is thus credited with enabling users to do their jobs better, especially faster. Interestingly, we see a similar pattern in questions Q18 and Q19, which also ask about the quality and quantity of work. Here, too, improvement in the quantity of work is rated slightly more positively than improvement in quality. Overall, our DPs are considered effective. Questions Q20-Q22 address the effectiveness criterion again, focusing on a company using a data catalog with our design. This group of questions received the most negative responses compared to the rest of the questions. Q21 and Q22 were each answered twice with *disagree*. Q20 and Q21 were each answered three times with *neither agree nor disagree*. This could be because, while participants see data catalogs as quite an important tool, a better-functioning data catalog has little impact on a company's reputation or job morale. There are more important factors influencing this. According to the participants, a data catalog nevertheless impacts a company's ability to innovate. A data catalog implementing our DPs will enhance this innovation ability.

In summary, the reusability of the DPs can be regarded as given. The questions were answered positively throughout. Nevertheless, some aspects cannot be explained without further ado. This is because the questionnaire design does not provide detailed insight into individual DPs.

7.3.2 Comparison of the Questionnaire and the Expert Discussions

This section compares the results of the questionnaire with those of the expert discussions. We want to determine whether the statements about the five criteria according to Iivari et al. are consistent. This allows us to identify whether participants were more sympathetic or critical in the questionnaire or the interviews. If the criteria values are similar, we can assume that the experts are broadly consistent in their statements. This outcome would make the participants' opinions about the criteria more significant.

Two visualizations were generated to compare the criteria across all DPs. The first Figure 7.10 aggregates the questions from the questionnaire according to the criteria of Iivari

et al. The response options *agree* and *strongly agree* were aggregated to *Positive*. Similarly, *disagree* and *strongly disagree* were combined into *Negative*. *Neither agree nor disagree* was transferred to *Neither positive nor negative*. Percentage values of the distribution are given to allow comparison with the statements from the expert discussions. The second

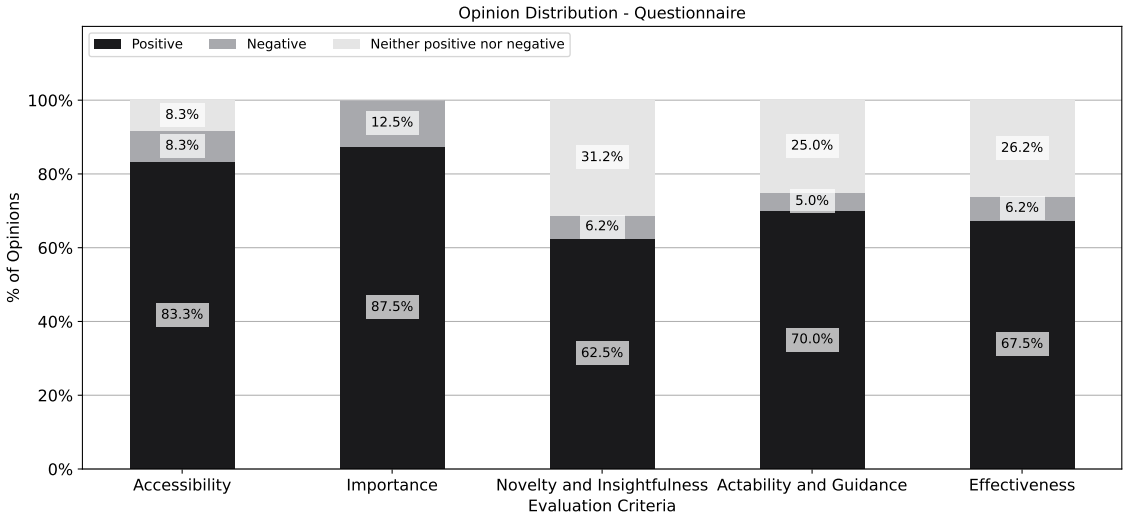


Figure 7.10: Aggregated questionnaire results

Figure 7.11 summarizes all previous evaluations from the expert discussions in one figure. Likewise, the distribution of opinions on the criteria was converted into percentage values. Due to the small number of eight participants, the comparison is exclusively descriptive.

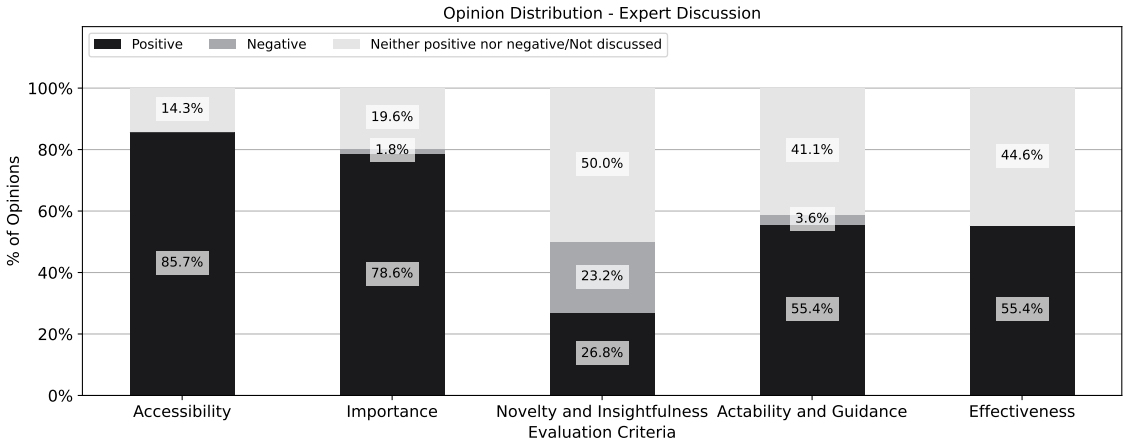


Figure 7.11: Aggregated expert discussion results

As can be seen from the figures, a somewhat similar picture is presented for both at first glance. The first criterion, **accessibility**, was mentioned positively by $\sim 83\%$ in the questionnaire and $\sim 86\%$ in the conversations. While there were no negative mentions in the conversations, there were $\sim 8\%$ negative mentions in the questionnaire. The numbers suggest that the experts across the different evaluation methods consistently rated the

accessibility of the DPs. Thus, at this point, we can say with a high degree of certainty that the totality of the DPs is formulated in a way that is understandable to the target group.

The second criterion, **importance**, was rated positively by $\sim 88\%$ in the questionnaire and $\sim 79\%$ in the discussions. While the remaining votes are exclusively negative in the questionnaire, they are predominantly non-judgmental in the discussions. The difference of the positive votes is $< 10\%$. Therefore, it can be assumed that the participants also consistently respond to this criterion. For this reason, and the very positive response, the totality of the DPs can be identified as important in their entirety.

The evaluations diverge for the third criterion, **novelty and insightfulness**. The questionnaire has $\sim 63\%$ positive votes and $\sim 31\%$ neutral votes. The negative votes hardly carry any weight at $\sim 6\%$. In the discussions, the topic of novelty across the DPs was only viewed positively by $\sim 27\%$ and negatively by $\sim 23\%$. Half do not take a position. This large discrepancy between the two evaluation methods can be explained by the fact that the questionnaire only asks questions about the totality of the DPs. The individual interviews also looked at the DPs individually. This allowed participants to comment specifically on individual DPs. The results indicate that no uniform picture of novelty can be formed about the totality of the DPs. As already listed in the evaluation of the individual DPs, there are principles that the participants classified as more novel and as not novel. For this criterion, there is no consistent statement between the two evaluation methods; thus, one cannot make a reliable statement about the totality of the DPs.

The fourth criterion, **actability and guidance**, was rated 70% positive, 25% neutral, and 5% negative in the questionnaire. In the discussions, the criterion was rated $\sim 55\%$ positive, $\sim 41\%$ neutral, and $\sim 4\%$ negative. Including the decimal place, the difference in positive ratings between the survey and the interviews is $< 15\%$. For the most part, the difference translates into neutral values. Due to the small number of negative comments and the fact that neutral values can also result from not being mentioned in the expert discussions, we can confidently expect a consistent response. Thus, an absolute majority considers the DPs to be implementable and contain sufficient information.

The fifth and thus last criterion of **effectiveness** was rated $\sim 68\%$ positive, $\sim 26\%$ neutral, and $\sim 6\%$ negative in the questionnaire. The criterion was rated $\sim 55\%$ positive in the discussions and $\sim 45\%$ neutral. There were no negative voices in the discussions. Since the difference between the positive evaluations is also $< 15\%$ here, we can assume a consistent response by the participants, as with the previous criterion. The absolute majority of the participants see the effectiveness of the DPs for data catalogs as given.

In summary, the participants gave predominantly consistent answers to the five criteria according to Iivari et al. The exception here is the novelty of the DPs. This may be explained by the fact that there is a great diversity in the novelty of individual DPs. Nevertheless, we can conclude here that the reusability of the DPs for data catalogs is given.

7.3.3 Discussion of Possible Deficiencies

We now discuss possible deficiencies according to Janiesch et al. [Jan20]. In the expert discussions, the participants were explicitly asked about the four types of deficiencies. First,

deficits were pointed out by two participants. E1 claimed that he lacks the explicit mention of legal aspects, such as licenses or GDPR, in the Principle of Context or Principle of Data Life Cycle Management. Participant E3 misses information in Principle of Flexibility that one should be able to connect new data sources to the data catalog easily. These comments are legitimate and should be considered in future iterations. However, both aspects refer to possible new or more concrete DFs. Our concretization of the DPs with the restriction by DIVA did not provide these aspects for DFs since legal aspects were only very weakly focused. For example, the practitioners wanted something like the specification of licenses and whether Personally Identifiable Information (PII) data is included. However, licenses are already part of data model standards such as DCAT, and the indication of whether PII is included can be added individually by an extension of the data model. Therefore, these are not principle deficits.

Participant E8 suggests a possible principle redundancy. Thus, according to E8, the Principle of Automation and the Principle of Interoperability aim to enable a data catalog that is complete in terms of content. These two DPs indeed complement each other well. Participants E2 and E7 agreed on this. However, the principles address more than just the content completeness of the catalog. For example, the Principle of Automation also strongly focuses on accurate, error-free, high-quality content. The Principle of Interoperability, on the other hand, focuses on integration into existing system landscapes and has no influence on accurate, error-free, and high-quality content for the time being. Due to this differentiation and the technical differences, which E8 also recognizes, the DPs should not be combined.

Finally, a possible principle excess was proposed for the Principle of Visualization. According to E8, there could be constellations in which a data catalog no longer has user interaction but is used entirely autonomously. In this case, visualizations for a faster and better understanding of the metadata by the user would no longer be a relevant phenomenon. E2 also says that a data catalog without GUI does not need visualizations. Participant E5 counters that even with such data catalogs, there will come a point when a human actor must intervene. Here, suitable visualizations should be offered at the latest.

In the course of the evaluation, three possible deficiencies could be identified. However, these were unrelated to DPs or could be invalidated by argumentation. In this respect, we have identified suitable DPs for data catalogs.

7.3.4 Overarching Expert Comments

One aspect expressed by E2, E5, and E7 is that the DPs can be implemented independently. The DPs overlap slightly, covering all essential aspects of data catalogs. The slight overlap, if there is any, also means that there are no gaps in this case (E7). The DPs are implemented to complement each other well (E5). However, one has to be careful at this point with the implementation of the Principle of Automation and Principle of Flexibility. These may push against each other, and one will not be able to achieve complete automation and flexibility (E8). The same holds for the Principle of Interoperability and Principle of Flexibility. Here, a compromise is needed, which has to be worked out with the data catalog users.

Another point mentioned by E2 and E4 relates to further detailing the concrete imple-

mentation. For example, the DFs are still very abstract for practitioners, and concrete implementation instructions are desired. This can be done, for example, in the form of decision trees (E2) or templates (E4).

Regarding the novelty of the DPs, E7 and E8 commented more generally. They see the DPs as lived practice, but this collection and listing are new. In this way, the DPs can be better remembered and refresh one's memory.

Finally, E4 points out the need for metrics to measure whether the individual DPs have been implemented. E4 acknowledges that this is not easy at such an abstract level. Nevertheless, at lower levels, at the latest, one should have such metrics to check whether the implementation of individual DFs was successful.

7.4 Discussion

First, it can be said that the evaluation paints a positive picture of the DPs for data catalogs. Thus, about the totality of the DPs, the five criteria according to Iivari et al. [Iiv21] were evaluated positively. This applies to both the questionnaire and the individual conversations with the experts. The positive evaluation in this case states that the DPs are reusable. They are useful and relevant in the implementation of data catalogs. Also, no deficiencies [Jan20] could be identified. Thus, in conclusion, we can say that we have developed a sound and complete set of DPs.

We still need to detail some aspects that emerge from the evaluation. First, we must note that the DFs were not evaluated extensively. They were added to the informational material but were only addressed occasionally. A more detailed evaluation of the DFs would be useful, especially because they are relevant for practitioners.

Another aspect to be discussed is the rather small number of participants. With eight participants, making reliable statements about the entire population is difficult. Also, statistical evaluation methods are not reasonably applicable here. For this reason, we conducted the entire evaluation descriptively. When selecting the participants, we ensured they covered the seven data catalog types discussed in Section 2.2.3. Therefore, we can assume that the DPs have been successfully decontextualized and apply to a wide range of data catalogs.

CHAPTER 8

Conclusion

Data catalogs are an essential tool in modern data management [Jah23; Lab20b; Sha16c]. By storing metadata about data and making it searchable for users, they facilitate data discovery [Ole23] and link data demand and supply [Jah23]. This also allows them to support the implementation of the FAIR principles [Bor22; Lab20b; Qui20], promote data democratization [Eic22a; Lef21] and contribute to the implementation of data governance [Che22; Rya22; Sha16c]. Supporting these three concepts is critical for enabling data-driven decision-making and gaining a competitive advantage [Leg17; Sam22; Szu23].

Until now, there has been a lack of comprehensive, practical-based, and design-oriented knowledge about data catalogs. We have closed this gap in the form of a design theory for data catalogs by extracting the necessary design knowledge from our data catalog DIVA. In the following, we will first provide concrete answers to our research questions and then summarize our contributions. Then, we will discuss our work's limitations before finally giving an outlook on future research opportunities.

8.1 Answers to the Research Questions

In Section 1.2, we formulated a series of research questions we now answer. We will briefly describe our approach and summarize our contributions. An overview of the design knowledge developed in this thesis can be found in Figure 8.1.

RQ1: *What are design principles for data catalogs?*
Answer to RQ1: To answer this research question, we followed a reflective development path [Möl20]. DPs were extracted from the data catalog software DIVA, which we developed in close collaboration with practitioners. This ensures our design knowledge has practical and research impact [Bas18a]. Care was taken to achieve a high level of abstraction to eliminate possible contextual idiosyncrasies of DIVA [Eng20; Gre13b]. This guarantees that our design-oriented knowledge approaches data catalogs holistically and that the DPs apply to all types of data catalogs [Kru16].

In total, we were able to identify **seven DPs** that are relevant for the implementation of data catalogs (see the left side of Figure 8.1). Practitioners can use this knowledge to implement successful instantiations of data catalogs. A standardized formulation of the DPs makes them easier to reuse in practice and research [Gre20]. We also support our DPs with literature from the data catalog knowledge base, thus ensuring practitioners and researchers can get further theoretical insights. Practitioners can utilize this knowledge to better comprehend the work that has already been done in the field of data catalogs related to the DP and then implement it. Researchers can expand on these studies and

do further research to support the DPs. We ensured that the DPs are free of defects and reusable by evaluating them with practitioners from the target group of this work [Iiv21; Jan20; Rec11]. We, therefore, have a verified set of DPs that can be considered useful for practitioners and researchers.

RQ2: *What are design features that support the concrete implementation of our design principles?*
Answer to RQ2: To answer this question, the DPs for data catalogs were concretized. In doing so, it was important not to implement just any concretizations but rather ones that correspond to our experience with DIVA development. Therefore, only those concretizations were implemented as DFs that are also reflected in DIVA.

In total, we were able to identify **18 DFs** that can support the implementation of our DPs (see the center of Figure 8.1). In order to support practitioners in implementing the DFs, we have enriched them with our experience gained in developing DIVA. On the one hand, we discussed each DF in detail to justify its existence and usefulness. On the other hand, we provide specific instructions on how the DFs can be implemented. In particular, we draw on specific features we have implemented in the context of DIVA. The DFs are particularly aimed at practitioners who want to implement our DPs. Due to the application examples, practitioners can gain further insights on a very concrete level into how DFs and thus DPs can be implemented. Some DFs are also supplemented by more concrete artifacts that we published (see right side of Figure 8.1). These publications can also provide practitioners with further insight into possible implementations of DFs.

First, we developed a metadata **model** that describes data as an asset [Spi18, Paper I.]. This model is based on DCAT and can be used as a database schema for the data catalog. In this way, we support compatibility with other data catalogs (DF3.1 Standardized Metadata Models) as well as the enrichment of important context knowledge (DF4.1 Metadata With Contextual Information), which is required to implement FAIR principles, data democratization, and data governance.

Second, we developed a **method** that allows users of data catalogs to develop their own metrics [Teb18, Paper II.]. Users can generate metrics based on existing metadata and create individual metrics for specific data types, projects or similar. Metrics are an important tool for evaluating data and should be individually adapted to the circumstances and needs of the user (DF2.2 Creation and Customization of Metrics). The system can be used to identify risks about data (DF7.3 Risk Management Capabilities) and helps to automatically calculate important key performance indicators (DF1.2 Automated Metadata Gathering).

Third, we designed an **architecture** that enables the distribution of data profiling tasks across a network of computers, utilizing existing processing capabilities at the edge while ensuring data sovereignty by only sharing data with machines that one trusts [Teb20, Paper III.]. The architecture is based on combining Kubernetes and Argo to perform the orchestration tasks and a P2P network in which data and code are exchanged between the peers. The flexible creation of workflows and use of a tag system allows users to execute tasks on machines of their choice (DF2.4 Configurable Workflow Engine). The architecture also supports the automatic collection of metadata by providing ready-made workflows for

various profiling tasks and then triggering them through the data catalog in the network (DF1.2 Automated Metadata Gathering).

Fourth, we developed the so-called Destroy Claim **model** and four **architecture** proposals for its integration in the data and technology landscape [Teb23d, Paper VII.]. The model allows the end of the data life cycle to be modeled in a standardized way (DF3.1 Standardized Metadata Models). This can automatically prevent data from continuing to be used due to poor quality, errors, or regulatory reasons (DF5.2 End-of-Life Data Management). The model and architecture were implemented in DIVA, and evaluation suggested that it is a reasonable approach that other data catalog systems should adopt.

In addition to the above, the following contributions were developed to generate the Destroy Claim model and architecture. First, we developed a data engineering reference model, which describes the important phases and perspectives in data engineering and provides information on which aspects receive little attention in the literature [Teb21, Paper IV.]. This work is supplemented by a survey of practitioners in which we determined the maturity level of the various phases in practice [Teb23a, Paper V.]. Both studies concluded that data deletion, and thus the end of the data life cycle in general, receives little to no attention in theory and practice. For this reason, we have analyzed the field of data deletion using an SLR and developed a taxonomy of data deletion [Teb23c, Paper VI.]. This, in turn, was the basis on which the Destroy Claim model and the architecture for integrating the Destroy Claims were developed [Teb23d, Paper VII.]. We also provide software as a contribution that can be used to generate Destroy Claim Agents, which can interpret Destroy Claims and execute them.

MRQ: *How to design data catalogs?*
Answer to MRQ: Until now, there has been a lack of holistic, practice-based, design-oriented knowledge for data catalogs in the form of a design theory (see related work in Chapter 3). This work is intended to close the gap. The eight components of a design theory according to Jones and Gregor [Jon07] were developed. We focused on developing the *Principles of Form and Function* and *Principles of Implementation*. We populated these components with practice-based DPs (RQ1) and DFs (RQ2). Our software artifact DIVA, which contains practical insights and implicit design knowledge, serves as an *Expository Instantiation*. In addition to DIVA, the DAC and the Destroy Claim Agent can also be found here. The other components were developed in the course of the work (*Purpose and Scope* and *Constructs*) or resulted directly from the context in which the work is positioned (*Artifact Mutability*, *Testable Propositions*, and *Justificatory Knowledge*).

Since the design theory consists of the contributions described above, the same implications apply. We mainly abstract implicit design knowledge, which contributes to the growing research efforts in the field of data catalogs. We offer a foundation upon which more in-depth knowledge regarding data catalogs can be investigated. As a result, further artifacts that validate our design theory can be developed using it as a starting point. Despite its large scope, our design theory is applicable to data governance, data democratization, and FAIR principles research. Researchers can use it to create specific, workable data catalog features that advance the three previously outlined concepts and, thus, their research streams. With our design theory, we also contribute to the data

management research field since data catalogs are essential to the discipline.

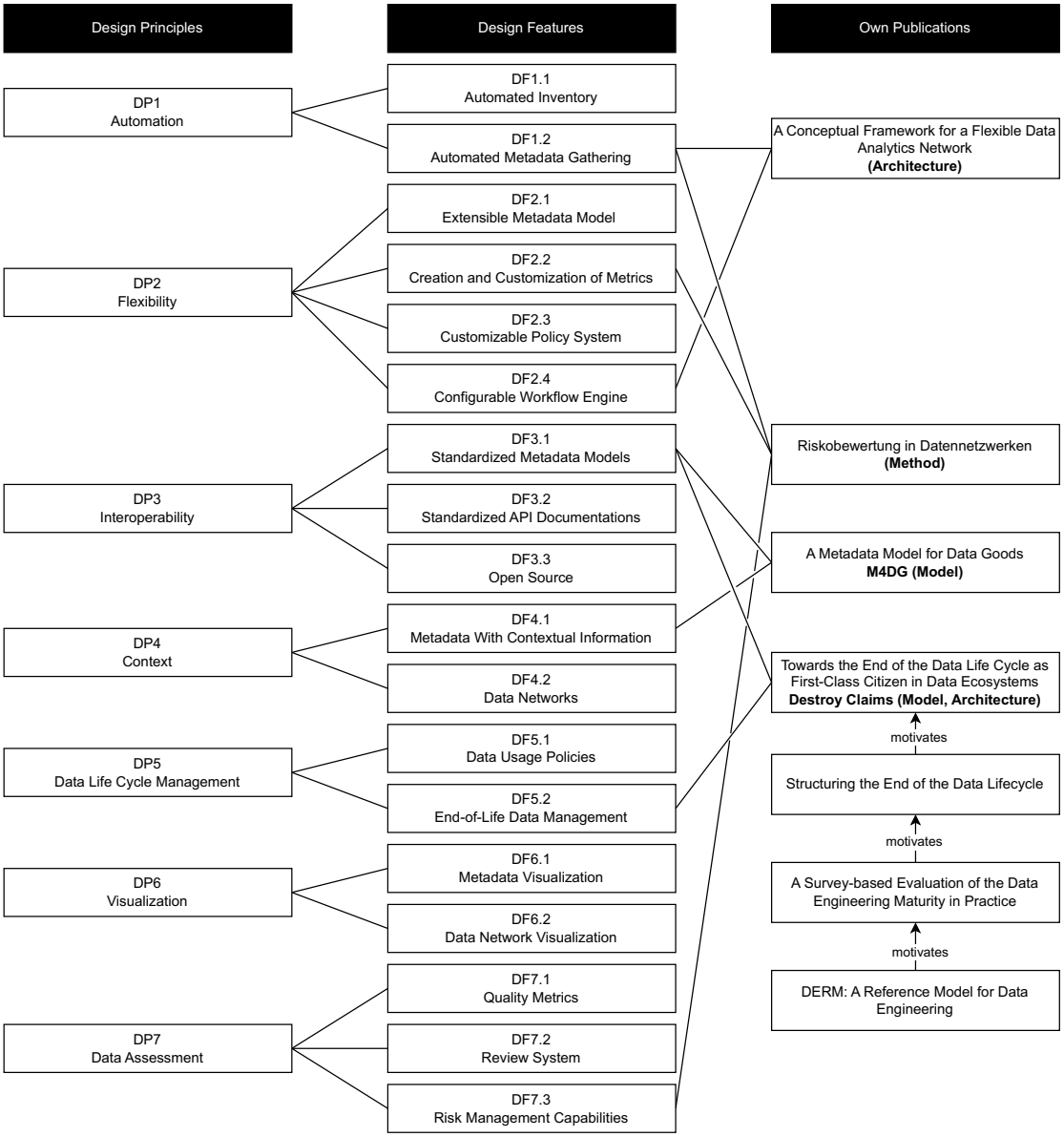


Figure 8.1: Overview of the mapping of DPs, DFs, and peer-reviewed artifacts

8.2 Limitations

Several aspects may limit the validity of the results of this work. First, the development of DIVA should be mentioned here. The data catalog software was adapted to individual needs in close exchange with practitioners. Despite the long development time of over six years, only eight iterations were carried out, incorporating new design knowledge. Of the eight iterations, five are associated with the same pharmaceutical company. It can, therefore, be

assumed that DIVA has a distinct domain bias. To mitigate this effect, the DPs have been abstracted, so domain constraints should no longer be relevant. Nevertheless, DIVA still presents a limitation in terms of DFs. In the context of this work, only those DFs were listed in the DPs, which are also reflected in DIVA. This constraint was done to ensure the relevance of the DFs. However, this has the consequence that not all relevant DFs have been identified. In DIVA, some relevant DFs are most likely missing due to the domain bias.

While we explicitly aimed to generalize practice-based knowledge, the question arises whether the reflective approach is suitable for identifying design knowledge in our case. We see creating a software artifact and extracting DPs as a sensible approach. "[It] can be strongly argued and defended that design of the IT artifact precedes the development of nascent design theories as a natural sequence of activities in a DSR project. While both activities are important, building and evaluating the artifact often comes first. Once the IT artifact is realized and evaluated in context, then the researchers have time to reflect and generate DPs for broader impacts of the embedded artifact knowledge to a wider range of applications." [Bas18a, pp. 362-363]. The fact that we used a single data catalog that was not created with the intention of extracting a design theory as our foundation may need to be revised to ensure the validity of our findings. We selected this single data catalog because we developed it ourselves and have the best insights into this software artifact. Developing the software in the first place to extract the design theory would also have induced a bias right from the start, as we would undoubtedly incorporate certain ideas. Nevertheless, this decision may have resulted in us not being able to identify all relevant DPs.

The supporting literature for our DPs was identified through an SLR. Whether an article was relevant and included in the corpus was decided by reading the abstract. This process has two potential shortcomings. The first is that the abstract may not reflect context, methods, goals, and results contrary to our assumption. The second is that we may have missed something while reading the abstract, and thus, articles were unjustifiably excluded. Therefore, the compilation of supporting literature should not be considered complete. The allocation to the DPs was also subject to our assessment, which means that literature may not be listed under all relevant DPs.

The next question is whether the process for identifying DPs can identify relevant ones. The DPs were identified through a process of reflection. Here, a bias that favors certain formulations and characteristics in the DPs can occur. To minimize this bias, the extraction of the DPs is followed by an evaluation.

For the evaluation, the question arises whether the applied methods are suitable. First, it should be noted that the evaluation was centered on the DPs rather than the design theory as a whole. This is because DPs are considered the most crucial part of a design theory [Hei14], and many works in the field also focus on them [Bit16; Cir14; Han17; Her22; Sch18a; Tuu18; Wag17]. The evaluation of the DPs was not done on a new data catalog that implements them. This type of evaluation is arguably the best because it evaluates the design naturally. However, developing a new data catalog that implements the design is complicated and time-consuming, so we decided against this method. Therefore, an artificial evaluation method was used. "To the extent that an artificial evaluation setting is

unreal, evaluation results may not correspond to real use." [Ven16, p. 81]. Therefore, the evaluation results are only partially transferable, even though they give strong guidance.

Two basic tests were performed during the evaluation. One was the reusability test, according to Iivari et al. [Iiv21], and the other was the deficiency test, according to Recker et al. [Rec11] and Janiesch et al. [Jan20]. For both tests, participants were chosen from the target group population. First, it can be said that the small number of eight participants can be a problem. Perhaps the population of the target group cannot be represented correctly. To reduce this effect, participants with experience with all seven types of data catalogs were selected. This guarantees that the design theory covers the complete class of data catalogs.

For the reusability test, a survey was conducted, which was completed by the participants alone. In an expert discussion, the participants were again confronted with the reusability criteria to be tested. By comparing the results, an overall consistency of the answers could be found. Thus, we can assume that the statements made in the one-to-one interviews were not, or only slightly, influenced by us and our way of asking. As a result, we can also assume that the participants answered truthfully concerning the individual statements on the DPs. Nevertheless, this is a qualitative evaluation that is subject to interpretation by us. The same applies to the deficiency assessment carried out in the one-to-one interviews.

Another area for improvement in the evaluation methodology concerns the DFs. These were handed out to participants along with the DPs. However, the survey only addresses the totality of DPs. Thus, one cannot draw any conclusions about the DFs here. During the expert discussions, the DFs were only occasionally discussed in passing. We assume that the participants accepted the DFs tacitly. Nevertheless, one should be aware that no reliable statements about the DFs exist.

Another limitation of the design theory becomes apparent in the context of its instantiation. Due to the high degree of abstraction of the DPs, own knowledge must be added by the implementers [Kru16]. This leads to the fact that a successful design of data catalogs cannot be guaranteed. The implementers could, for example, make wrong assumptions during the instantiation process and thus incorrectly shape the design theory in detail. Thus, no general success of the design theory can be guaranteed during the instantiation process. The transferability of our results beyond our study heavily depends on those who implement and supplement the design knowledge with their own knowledge.

Finally, whether our findings can be applied to a context that extends beyond our work arises. We have construed the DPs, and thus the design theory, for all types of data catalogs. The evaluation suggests that this goal has been achieved. Nevertheless, the design theory cannot be applied arbitrarily to other types of software. The next higher class of systems above a data catalog would be metadata management tools. It is already questionable whether applicability to the generality of metadata management tools is feasible. Metadata management tools can fill particular niches and have distinctive features. It is easy to imagine that a self-contained system for metadata management, for example, is based only on automation and otherwise does not have to implement any of the other principles. Therefore, it can be said that the results can only be applied to the data catalogs covered by our design theory (see Section 6.8).

8.3 Future Work

Several aspects might be addressed to improve and strengthen our design theory for data catalogs. First, the individual components of the design theory must be repeatedly confirmed in theory and practice. This can provide more evidence that the design theory is correct. The confirmation and settling of these components can be done, for example, in the context of re-evaluating the DPs. Here, a naturalistic evaluation should be pursued, in which one or more new data catalogs are developed that implement our design knowledge. Based on these new data catalogs, the DPs can be strengthened, or new findings can be incorporated. The data catalogs developed in this way can also serve as further *Expository Instantiations*.

Second, another task would be complementing and refining the DFs. As the previous section shows, the DFs are not a completed set but are grounded by the data catalog DIVA. For the sake of practitioners and the possible implementation of new data catalogs, it would be helpful to extend the DFs. For example, the expansion can be done by analyzing other existing data catalogs. Another possibility is the analysis of existing publications. This thesis has already created a corpus of data catalogs, and the contained articles have been mapped to the DPs. These articles can be used again to extract DFs.

Third, participants in the evaluation discussed the refinement of the DFs to an even more concrete level. Practical templates and decision support were mentioned. In future work, very concrete solutions can also be developed as ready-made libraries. In particular, more complex topics such as data profiling or communication between data catalogs and other data-holding systems are potential starting points here.

Bibliography

[Abe15] Abedjan, Ziawasch, Lukasz Golab, and Felix Naumann: 'Profiling relational data: a survey'. *VLDB J.* (2015), vol. 24(4): pp. 557–581 (cit. on p. 16).

[Abr19] Abraham, Rene, Johannes Schneider, and Jan Vom Brocke: 'Data governance: A conceptual framework, structured review, and research agenda'. *International journal of information management* (2019), vol. 49: pp. 424–438 (cit. on p. 2).

[Act96] Act, Accountability: 'Health insurance portability and accountability act of 1996'. *Public law* (1996), vol. 104: p. 191 (cit. on p. 8).

[Afg22] Afgan, Enis et al.: 'The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2022 update'. *Nucleic Acids Res.* (2022), vol. 50(W1): pp. 345–351 (cit. on pp. 52, 59).

[Ake04] Aken, Joan E van: 'Management research based on the paradigm of the design sciences: the quest for field-tested and grounded technological rules'. *Journal of management studies* (2004), vol. 41(2): pp. 219–246 (cit. on pp. 4, 28).

[Ala09] Alasem, Abdurrahman: 'An Overview of e-Government Metadata Standards and Initiatives based on Dublin Core'. *Electronic Journal of e-Government* (2009), vol. 7(1): pp1–10 (cit. on p. 9).

[Alb23] Albertoni, Riccardo, David Browning, Simon Cox, Alejandra N Gonzalez-Beltran, Andrea Perego, and Peter Winstanley: 'The W3C Data Catalog Vocabulary, Version 2: Rationale, Design Principles, and Uptake'. *arXiv preprint arXiv:2303.08883* (2023), vol. (cit. on pp. 9, 62).

[Ale09] Alexander, Keith, Richard Cyganiak, Michael Hausenblas, and Jun Zhao: 'Describing Linked Datasets'. *Proceedings of the WWW2009 Workshop on Linked Data on the Web, LDOW 2009, Madrid, Spain, April 20, 2009.* Ed. by Bizer, Christian, Tom Heath, Tim Berners-Lee, and Kingsley Idehen. Vol. 538. CEUR Workshop Proceedings. CEUR-WS.org, 2009 (cit. on p. 9).

[Ale11] Alexander, Keith, Richard Cyganiak, Michael Hausenblas, and Jun Zhao: 'Describing linked datasets with the VoID vocabulary'. (2011), vol. (cit. on p. 8).

[Ano20] Anouncia, S Margret, Hardik A Gohel, and Subbiah Vairamuthu: *Data Visualization.* Springer, 2020 (cit. on p. 73).

[Apa15] Aparicio, Manuela and Carlos J Costa: 'Data visualization'. *Communication design quarterly review* (2015), vol. 3(1): pp. 7–11 (cit. on p. 73).

[Arm02] Arms, William Y., Diane Hillmann, Carl Lagoze, Dean B. Krafft, Richard J. Marisa, John Saylor, Carol Terrizzi, and Herbert Van de Sompel: 'A Spectrum of Interoperability: The Site for Science Prototype for the NSDL'. *D Lib Mag.* (2002), vol. 8(1) (cit. on p. 61).

[Awa20] Awasthi, Pranjal and Jordana J. George: 'A case for Data Democratization'. *26th Americas Conference on Information Systems, AMCIS 2020, Virtual Conference, August 15-17, 2020*. Ed. by Anderson, Bonnie Brinton, Jason Thatcher, Rayman D. Meservy, Kathy Chudoba, Kelly J. Fadel, and Sue Brown. Association for Information Systems, 2020 (cit. on p. 2).

[Axt16] Axton, Myles, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, Jildau Bouwman, Anthony J Brookes, Tim Clark, M Crosas, et al.: 'The FAIR Guiding Principles for scientific data management and stewardship'. *Scientific data* (2016), vol. 3(1) (cit. on p. 2).

[Azc22] Azcoitia, Santiago Andrés and Nikolaos Laoutaris: 'A Survey of Data Marketplaces and Their Business Models'. *SIGMOD Rec.* (2022), vol. 51(3): pp. 18–29 (cit. on pp. 1, 14).

[Bad20] Bader, Sebastian R., Jaroslav Pullmann, Christian Mader, Sebastian Tramp, Christoph Quix, Andreas W. Müller, Haydar Akyürek, Matthias Böckmann, Benedikt T. Imbusch, Johannes Lipp, Sandra Geisler, and Christoph Lange: 'The International Data Spaces Information Model - An Ontology for Sovereign Exchange of Digital Content'. *The Semantic Web - ISWC 2020 - 19th International Semantic Web Conference, Athens, Greece, November 2-6, 2020, Proceedings, Part II*. Ed. by Pan, Jeff Z., Valentina A. M. Tamma, Claudia d'Amato, Krzysztof Janowicz, Bo Fu, Axel Polleres, Oshani Seneviratne, and Lalana Kagal. Vol. 12507. Lecture Notes in Computer Science. Springer, 2020: pp. 176–192 (cit. on p. 63).

[Bar22a] Barcellos, Monalessa, Gleison Santos, Tayana Conte, Bianca Trinkenreich, and Patrícia Matsubara: 'Organizing Empirical Studies as Learning Iterations in Design Science Research Projects'. *Proceedings of the XXI Brazilian Symposium on Software Quality, SBQS 2022, Curitiba, Brazil, November 7-10, 2022*. Ed. by Canedo, Edna Dias et al. ACM, 2022: 17:1–17:10 (cit. on p. 23).

[Bas18a] Baskerville, Richard L., Abayomi Baiyere, Shirley Gregor, Alan R. Hevner, and Matti Rossi: 'Design Science Research Contributions: Finding a Balance between Artifact and Theory'. *J. Assoc. Inf. Syst.* (2018), vol. 19(5): p. 3 (cit. on pp. 26, 105, 109).

[Bas16] Basole, Rahul C., Jukka Huhtamäki, Kaisa Still, and Martha G. Russell: 'Visual decision support for business ecosystem analysis'. *Expert Syst. Appl.* (2016), vol. 65: pp. 271–282 (cit. on p. 73).

[Bas09] Bastian, Mathieu, Sebastien Heymann, and Mathieu Jacomy: 'Gephi: An Open Source Software for Exploring and Manipulating Networks'. *Proceedings of the Third International Conference on Weblogs and Social Media, ICWSM 2009, San Jose, California, USA, May 17-20, 2009*. Ed. by Adar, Eytan, Matthew Hurst, Tim Finin, Natalie S. Glance, Nicolas Nicolov, and Belle L. Tseng. The AAAI Press, 2009 (cit. on p. 75).

[Ber09] Berthold, Michael R., Nicolas Cebron, Fabian Dill, Thomas R. Gabriel, Tobias Kötter, Thorsten Meinl, Peter Ohl, Kilian Thiel, and Bernd Wiswedel: 'KNIME - the Konstanz information miner: version 2.0 and beyond'. *SIGKDD Explor.* (2009), vol. 11(1): pp. 26–31 (cit. on pp. 52, 59).

[Bis16] Biswas, Prosunjit, Ravi S. Sandhu, and Ram Krishnan: 'An Attribute-Based Protection Model for JSON Documents'. *Network and System Security - 10th International Conference, NSS 2016, Taipei, Taiwan, September 28-30, 2016, Proceedings*. Ed. by Chen, Jiageng, Vincenzo Piuri, Chunhua Su, and Moti Yung. Vol. 9955. Lecture Notes in Computer Science. Springer, 2016: pp. 303–317 (cit. on p. 58).

[Bit16] Bitzer, Philipp, Matthias Söllner, and Jan Marco Leimeister: 'Design Principles for High-Performance Blended Learning Services Delivery - The Case of Software Trainings in Germany'. *Bus. Inf. Syst. Eng.* (2016), vol. 58(2): pp. 135–149 (cit. on p. 109).

[Boe18] Boeckhout, Martin, Gerhard A Zielhuis, and Annelien L Bredenoord: 'The FAIR guiding principles for data stewardship: fair enough?' *European journal of human genetics* (2018), vol. 26(7): pp. 931–936 (cit. on p. 2).

[Bor19] Boris Otto et al.: *Reference Architecture Model*. `https://www.fraunhofer.de/content/dam/zv/en/fields-of-research/industrial-data-space/IDS-Reference-Architecture-Model.pdf` [Accessed: August 15, 2023]. 2019 (cit. on pp. 37, 38).

[Bör03] Börner, Katy, Chaomei Chen, and Kevin W Boyack: 'Visualizing knowledge domains'. *Annual review of information science and technology* (2003), vol. 37(1): pp. 179–255 (cit. on p. 75).

[Bor18] Borrmann, André, Jakob Beetz, Christian Koch, Thomas Liebich, and Sergej Muhic: 'Industry foundation classes: A standardized data model for the vendor-neutral exchange of digital building models'. *Building information modeling: Technology foundations and industry practice* (2018), vol.: pp. 81–126 (cit. on p. 62).

[Bra17] Bratt, Sarah, Jeff Hemsley, Jian Qin, and Mark Costa: 'Big data, big metadata and quantitative study of science: A workflow model for big scientometrics'. *Proceedings of the Association for Information Science and Technology* (2017), vol. 54(1): pp. 36–45 (cit. on p. 3).

[Bra06] Braun, Virginia and Victoria Clarke: 'Using thematic analysis in psychology'. *Qualitative research in psychology* (2006), vol. 3(2): pp. 77–101 (cit. on p. 26).

[Bro03] Broekstra13, Jeen, Marc Ehrig, Peter Haase, Frank Van Harmelen, Arjohn Kampman, Marta Sabou, Ronny Siebes, Steffen Staab, Heiner Stuckenschmidt, and Christoph Tempich: 'A metadata model for semantics-based peer-to-peer systems'. *SemPGRID'03* (2003), vol. (cit. on p. 8).

[Bur19] Burnay, Corentin, Fátima C. C. Dargam, and Pascale Zaraté: 'Special issue: Data visualization for decision-making: an important issue'. *Oper. Res.* (2019), vol. 19(4): pp. 853–855 (cit. on p. 73).

[Cal24] Calderon-Monge, Esther and Domingo Ribeiro-Soriano: 'The role of digitalization in business and management: a systematic literature review'. *Review of managerial science* (2024), vol. 18(2): pp. 449–491 (cit. on p. 1).

[Car22] Carruthers, Andrew: 'Breaking Data Silos'. *Building the Snowflake Data Cloud: Monetizing and Democratizing Your Data*. Berkeley, CA: Apress, 2022: pp. 29–50 (cit. on p. 3).

[Caw07] Cawthon, Nick and Andrew Vande Moere: 'The effect of aesthetic on the usability of data visualization'. *2007 11th International Conference Information Visualization (IV'07)*. IEEE. 2007: pp. 637–648 (cit. on pp. 73, 74).

[Cha14] Chae, Bongsug Kevin, Chenlung Yang, David Olson, and Chwen Sheu: 'The impact of advanced analytics and data accuracy on operational performance: A contingent resource based theory (RBT) perspective'. *Decision support systems* (2014), vol. 59: pp. 119–126 (cit. on pp. 71, 76).

[Cha15a] Chaki, Saumya and Saumya Chaki: 'The lifecycle of enterprise information management'. *Enterprise Information Management in Practice: Managing Data and Leveraging Profits in Today's Complex Business Environment* (2015), vol.: pp. 7–14 (cit. on p. 69).

[Cha15b] Chandra, Leona, Stefan Seidel, and Shirley Gregor: 'Prescriptive knowledge in IS research: Conceptualizing design principles in terms of materiality, action, and boundary conditions'. *2015 48th hawaii international conference on system sciences*. IEEE. 2015: pp. 4039–4048 (cit. on pp. 4, 27).

[Cir14] Ciriello, Raffaele, Felix-Robinson Aschoff, Mateusz Dolata, and Alexander Richter: 'Communicating Ideas Purposefully - toward a Design Theory of Innovation Artifacts'. *22st European Conference on Information Systems, ECIS 2014, Tel Aviv, Israel, June 9-11, 2014*. Ed. by Avital, Michel, Jan Marco Leimeister, and Ulrike Schultze. 2014 (cit. on p. 109).

[Cou22] Council of the EU, General Secretariat of the Council: *Energy crisis: Three EU-coordinated measures to cut down bills*. 2022. URL: https://www.consilium.europa.eu/en/infographics/eu-measures-to-cut-down-energy-bills/ (visited on 10/24/2022) (cit. on p. 71).

[Cro18] Cronholm, Stefan and Hannes Göbel: 'Guidelines Supporting the Formulation of Design Principles'. *Australasian Conference on Information Systems, ACIS 2018, Sydney, NSW, Australia, 3-5 December 2018*. 2018: p. 8 (cit. on p. 27).

[Dem14] Demchenko, Yuri, Cees De Laat, and Peter Membrey: 'Defining architecture components of the Big Data Ecosystem'. *2014 International conference on collaboration technologies and systems (CTS)*. IEEE. 2014: pp. 104–112 (cit. on p. 69).

[Dem20] Demestichas, Konstantinos and Emmanouil Daskalakis: 'Data lifecycle management in precision agriculture supported by information and communication technology'. *Agronomy* (2020), vol. 10(11): p. 1648 (cit. on p. 69).

[Den17] Deng, Dong, Raul Castro Fernandez, Ziawasch Abedjan, Sibo Wang, Michael Stonebraker, Ahmed K. Elmagarmid, Ihab F. Ilyas, Samuel Madden, Mourad Ouzzani, and Nan Tang: 'The Data Civilizer System'. *8th Biennial Conference on Innovative Data Systems Research, CIDR 2017, Chaminade, CA, USA, January 8-11, 2017, Online Proceedings*. www.cidrdb.org, 2017 (cit. on p. 3).

[Di 94] Di Battista, Giuseppe, Peter Eades, Roberto Tamassia, and Ioannis G Tollis: 'Algorithms for drawing graphs: an annotated bibliography'. *Computational Geometry* (1994), vol. 4(5): pp. 235–282 (cit. on p. 75).

[Dia18] Diamantini, Claudia, Paolo Lo Giudice, Lorenzo Musarella, Domenico Potena, Emanuele Storti, and Domenico Ursino: 'A New Metadata Model to Uniformly Handle Heterogeneous Data Lake Sources'. *New Trends in Databases and Information Systems - ADBIS 2018 Short Papers and Workshops, AI*QA, BIGPMED, CSACDB, M2U, BigDataMAPS, ISTREND, DC, Budapest, Hungary, September, 2-5, 2018, Proceedings*. Ed. by Benczúr, András, Bernhard Thalheim, Tomás Horváth, Silvia Chiusano, Tania Cerquitelli, Csaba István Sidló, and Peter Z. Revesz. Vol. 909. Communications in Computer and Information Science. Springer, 2018: pp. 165–177 (cit. on p. 8).

[Ede06] Eden, Amnon H. and Tom Mens: 'Measuring software flexibility'. *IEE Proc. Softw.* (2006), vol. 153(3): pp. 113–125 (cit. on p. 55).

[Ehr21b] Ehrlinger, Lisa, Johannes Schrott, Martin Melichar, Nicolas Kirchmayr, and Wolfram Wöss: 'Data Catalogs: A Systematic Literature Review and Guidelines to Implementation'. *Database and Expert Systems Applications - DEXA 2021 Workshops - BIOKDD, IWCFS, MLKgraphs, AI-CARES, ProTime, AISys 2021, Virtual Event, September 27-30, 2021, Proceedings*. Ed. by

Kotsis, Gabriele, A Min Tjoa, Ismail Khalil, Bernhard Moser, Atif Mashkoor, Johannes Sametinger, Anna Fensel, Jorge Martínez Gil, Lukas Fischer, Gerald Czech, Florian Sobieczky, and Sohail Khan. Vol. 1479. Communications in Computer and Information Science. Springer, 2021: pp. 148–158 (cit. on pp. 3, 4).

[Eic21] Eichler, Rebecca, Corinna Giebler, Christoph Gröger, Eva Hoos, Holger Schwarz, and Bernhard Mitschang: 'Enterprise-Wide Metadata Management An Industry Case on the Current State and Challenges'. *24th International Conference on Business Information Systems, BIS 2021, Hannover, Germany, June 15-17, 2021*. Ed. by Abramowicz, Witold, Sören Auer, and Elzbieta Lewanska. 2021: pp. 269–279 (cit. on p. 3).

[Eic20] Eichler, Rebecca, Corinna Giebler, Christoph Gröger, Holger Schwarz, and Bernhard Mitschang: 'HANDLE - A Generic Metadata Model for Data Lakes'. *Big Data Analytics and Knowledge Discovery - 22nd International Conference, DaWaK 2020, Bratislava, Slovakia, September 14-17, 2020, Proceedings*. Ed. by Song, Min, Il-Yeol Song, Gabriele Kotsis, A Min Tjoa, and Ismail Khalil. Vol. 12393. Lecture Notes in Computer Science. Springer, 2020: pp. 73–88 (cit. on p. 8).

[Eic22c] Eichler, Rebecca, Christoph Gröger, Eva Hoos, Holger Schwarz, and Bernhard Mitschang: 'From Data Asset to Data Product–The Role of the Data Provider in the Enterprise Data Marketplace'. *Service-Oriented Computing: 16th Symposium and Summer School, SummerSOC 2022, Hersonissos, Crete, Greece, July 3–9, 2022, Revised Selected Papers*. Springer. 2022: pp. 119–138 (cit. on p. 14).

[Eit21] Eitel, Andreas, Christian Jung, Robin Brandstädter, Arghavan Hosseinzadeh, Sebastian Bader, Christian Kühnle, Pascal Birnstill, Gerd Brost, Mark Gall, Fabian Bruckner, Norbert Weissenberg, and Benjamin Korth: 'Usage Control in the International Data Spaces: Version 3.0'. (2021), vol. [Accessed April 29, 2022] (cit. on p. 71).

[Eng20] Engström, Emelie, Margaret-Anne D. Storey, Per Runeson, Martin Höst, and Maria Teresa Baldassarre: 'How software engineering research aligns with design science: a review'. *Empir. Softw. Eng.* (2020), vol. 25(4): pp. 2630–2660 (cit. on pp. 5, 24, 26, 105).

[16] *The European Data Portal: Opening up Europe's public data*. `https://data.europa.eu/sites/default/files/edp_factsheet_what_is_edp_project_online.pdf`. 2016 (cit. on p. 79).

[Eur16] European Commission: *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance)*. 2016 (cit. on pp. 8, 71).

[Fau13] Faundeen, John L, Thomas E Burley, Jennifer Carlino, David L Govoni, Heather S Henkel, Sally Holl, Vivian B Hutchison, Elizabeth Martín, Ellyn T Montgomery, Cassandra C Ladino, et al.: *The United States geological survey science data lifecycle model*. US Department of the Interior, US Geological Survey Reston, VA, USA, 2013 (cit. on p. 69).

[Few07] Few, Stephen and Perceptual Edge: 'Dashboard confusion revisited'. *Perceptual Edge* (2007), vol.: pp. 1–6 (cit. on p. 74).

[Fie91] Fiegenbaum, Avi and Aneel Karnani: 'Output flexibility—a competitive advantage for small firms'. *Strategic management journal* (1991), vol. 12(2): pp. 101–114 (cit. on p. 54).

[Flo17] Florescu, Corina and Cornelia Caragea: 'PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents'. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*. Ed. by Barzilay, Regina and Min-Yen Kan. Association for Computational Linguistics, 2017: pp. 1105–1115 (cit. on p. 34).

[Fra05b] Franklin, Michael J., Alon Y. Halevy, and David Maier: 'From databases to dataspaces: a new abstraction for information management'. *SIGMOD Rec.* (2005), vol. 34(4): pp. 27–33 (cit. on p. 11).

[Fre21] Freche, Jens, Milan den Heijer, and Bastian Wormuth: 'Data Lineage'. *The Digital Journey of Banking and Insurance, Volume III: Data Storage, Data Processing and Data Analysis* (2021), vol.: pp. 5–19 (cit. on p. 75).

[Fu15] Fu, Katherine K, Maria C Yang, and Kristin L Wood: 'Design principles: The foundation of design'. *International design engineering technical conferences and computers and information in engineering conference*. Vol. 57175. American Society of Mechanical Engineers. 2015: V007T06A034 (cit. on p. 5).

[Gam21] Gamal, Aya, Sherif I. Barakat, and Amira Rezk: 'Standardized electronic health record data modeling and persistence: A comparative review'. *J. Biomed. Informatics* (2021), vol. 114: p. 103670 (cit. on p. 62).

[Gao19] Gao, Biao, Bo Chen, Shijie Jia, and Luning Xia: 'eHIFS: An Efficient History Independent File System'. *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security, AsiaCCS 2019, Auckland, New Zealand, July 09-12, 2019*. Ed. by Galbraith, Steven D., Giovanni Russello, Willy Susilo, Dieter Gollmann, Engin Kirda, and Zhenkai Liang. ACM, 2019: pp. 573–585 (cit. on p. 71).

[Geb06] Gebauer, Judith and Franz Schober: 'Information System Flexibility and the Cost Efficiency of Business Processes'. *J. Assoc. Inf. Syst.* (2006), vol. 7(3): p. 8 (cit. on p. 55).

[Gou22] Gouert, Charles and Nektarios Georgios Tsoutsos: 'Dirty Metadata: Understanding A Threat to Online Privacy'. *IEEE Secur. Priv.* (2022), vol. 20(6): pp. 27–34 (cit. on pp. 3, 9).

[Gre09] Gregor, Shirley: 'Building theory in the sciences of the artificial'. *Proceedings of the 4th international conference on design science research in information systems and technology*. 2009: pp. 1–10 (cit. on pp. 25, 27).

[Gre02] Gregor, Shirley et al.: 'Design theory in information systems'. *Australasian Journal of Information Systems* (2002), vol. 10(1) (cit. on p. 4).

[Gre20] Gregor, Shirley, Leona Chandra Kruse, Stefan Seidel, et al.: 'Research perspectives: the anatomy of a design principle'. Association for Information Systems. 2020 (cit. on pp. 4, 27, 28, 105).

[Gre13a] Gregor, Shirley and Alan R. Hevner: 'Positioning and Presenting Design Science Research for Maximum Impact'. *MIS Q.* (2013), vol. 37(2): pp. 337–355 (cit. on pp. 4, 25, 82).

[Gre13b] Gregor, Shirley, Oliver Müller, and Stefan Seidel: 'Reflection, Abstraction And Theorizing In Design And Development Research.' *ECIS 2013 Completed Research*. Paper 74. 2013 (cit. on pp. 3, 5, 105).

[Hai22] Haines, Scott: 'Workflow Orchestration with Apache Airflow'. *Modern Data Engineering with Apache Spark: A Hands-On Guide for Building Mission-Critical Streaming Applications*. Berkeley, CA: Apress, 2022: pp. 255–295 (cit. on p. 59).

[Han17] Hansen, Magnus Rotvit Perlt and Jan Pries-Heje: 'Value Creation in Knowledge Networks. Five design principles'. *Scand. J. Inf. Syst.* (2017), vol. 29(2): p. 3 (cit. on p. 109).

[Hei14] Heinrich, Peter and Gerhard Schwabe: 'Communicating Nascent Design Theories on Innovative Information Systems through Multi-grounded Design Principles'. *Advancing the Impact of Design Science: Moving from Theory to Practice - 9th International Conference, DESRIST 2014, Miami, FL, USA, May 22-24, 2014. Proceedings*. Ed. by Tremblay, Monica Chiarini, Debra E. VanderMeer, Marcus A. Rothenberger, Ashish Gupta, and Victoria Y. Yoon. Vol. 8463. Lecture Notes in Computer Science. Springer, 2014: pp. 148–163 (cit. on pp. 4, 24, 85, 109).

[Her22] Herm, Lukas-Valentin, Theresa Steinbach, Jonas Wanner, and Christian Janiesch: 'A nascent design theory for explainable intelligent systems'. *Electron. Mark.* (2022), vol. 32(4): pp. 2185–2205 (cit. on p. 109).

[Her00] Herman, Ivan, Guy Melançon, and M Scott Marshall: 'Graph visualization and navigation in information visualization: A survey'. *IEEE Transactions on visualization and computer graphics* (2000), vol. 6(1): pp. 24–43 (cit. on p. 75).

[Hev04] Hevner, Alan R, Salvatore T March, Jinsoo Park, and Sudha Ram: 'Design science in information systems research'. *MIS quarterly* (2004), vol.: pp. 75–105 (cit. on pp. 23, 85).

[Hir22] Hirzel, Martin: 'Low-Code Programming Models'. *CoRR* (2022), vol. abs/2205.02282 (cit. on p. 57).

[Hub13] Hubert Ofner, Martin, Kevin Straub, Boris Otto, and Hubert Oesterle: 'Management of the master data lifecycle: a framework for analysis'. *Journal of Enterprise Information Management* (2013), vol. 26(4): pp. 472–491 (cit. on p. 69).

[Hus15] Huser, Vojtech, Chandan Sastry, Matthew Breymaier, Asma Idriss, and James J. Cimino: 'Standardizing data exchange for clinical research protocols and case report forms: An assessment of the suitability of the Clinical Data Interchange Standards Consortium (CDISC) Operational Data Model (ODM)'. *J. Biomed. Informatics* (2015), vol. 57: pp. 88–99 (cit. on p. 62).

[Iiv03] Iivari, Juhani: 'The IS core-VII: Towards information systems as a science of meta-artifacts'. *Communications of the Association for Information Systems* (2003), vol. 12(1): p. 37 (cit. on p. 4).

[Iiv18] Iivari, Juhani, Magnus Rotvit Perlt Hansen, and Amir Haj-Bolouri: 'A framework for light reusability evaluation of design principles in design science research'. *13th International Conference on Design Science Research and Information Systems and Technology: Designing for a Digital and Globalized World*. 2018: pp. 1–15 (cit. on pp. 85, 86, 88).

[Iiv21] Iivari, Juhani, Magnus Rotvit Perlt Hansen, and Amir Haj-Bolouri: 'A proposal for minimum reusability evaluation of design principles'. *Eur. J. Inf. Syst.* (2021), vol. 30(3): pp. 286–303 (cit. on pp. 24, 85, 86, 88, 89, 103, 106, 110).

[Int17] International, Dama: *DAMA-DMBOK: data management body of knowledge*. Technics Publications, LLC, 2017 (cit. on p. 2).

[Jac22] Jackson, Thomas W and Ian Richard Hodgkinson: 'Keeping a lower profile: how firms can reduce their digital carbon footprints'. *Journal of Business Strategy* (2022), vol. (ahead-of-print) (cit. on p. 71).

[Jac20] Jacobsen, Annika, Ricardo de Miranda Azevedo, Nick Juty, Dominique Batista, Simon Coles, Ronald Cornet, Mélanie Courtot, Mercè Crosas, Michel Dumontier, Chris T Evelo, et al.: *FAIR principles: interpretations and implementation considerations*. 2020 (cit. on p. 1).

[Jah23] Jahnke, Nils and Boris Otto: 'Data Catalogs in the Enterprise: Applications and Integration'. *Datenbank-Spektrum* (June 2023), vol. (cit. on pp. 1, 3, 4, 9, 10, 12–14, 51, 105).

[Jah22] Jahnke, Nils, Markus Spiekermann, and Behnam Ramouzeh: *Data Catalogs: Implementing Capabilities for Data Curation, Data Enablement and Regulatory Compliance - 2022 Edition*. `https://www.isst.fraunhofer.de/content/dam/isst-neu/documents/Publikationen/Datenwirtschaft/Fraunhofer-ISST_DataCatalogs_Report-kl.pdf`. [Accessed: March 30, 2023]. 2022 (cit. on p. 75).

[Jan20] Janiesch, Christian, Christoph Rosenkranz, and Ulrich Scholten: 'An Information Systems Design Theory for Service Network Effects'. *J. Assoc. Inf. Syst.* (2020), vol. 21(6): p. 9 (cit. on pp. 24, 85–87, 101, 103, 106, 110).

[Jef20] Jeffery, Keith G.: 'Data Curation and Preservation'. *Towards Interoperable Research Infrastructures for Environmental and Earth Sciences - A Reference Model Guided Approach for Common Challenges*. Ed. by Zhao, Zhiming and Margareta Hellström. Vol. 12003. Lecture Notes in Computer Science. Springer, 2020: pp. 123–139 (cit. on pp. 3, 7–9, 51).

[Jia17] Jiang, Hao and Ahmed Bouabdallah: 'JACPoL: A Simple but Expressive JSON-Based Access Control Policy Language'. *Information Security Theory and Practice - 11th IFIP WG 11.2 International Conference, WISTP 2017, Heraklion, Crete, Greece, September 28-29, 2017, Proceedings*. Ed. by Hancke, Gerhard P. and Ernesto Damiani. Vol. 10741. Lecture Notes in Computer Science. Springer, 2017: pp. 56–72 (cit. on p. 58).

[Jiv18] Jivet, Ioana, Maren Scheffel, Marcus Specht, and Hendrik Drachsler: 'License to evaluate: Preparing learning analytics dashboards for educational practice'. *Proceedings of the 8th international conference on learning analytics and knowledge*. 2018: pp. 31–40 (cit. on p. 74).

[Joh09] Johnson, Theodore: *Encyclopedia of Database Systems, chapter Data Profiling*. 2009 (cit. on p. 16).

[Jon07] Jones, David and Shirley Gregor: 'The Anatomy of a Design Theory'. *J. Assoc. Inf. Syst.* (2007), vol. 8(5): p. 19 (cit. on pp. 4, 5, 10, 23–25, 80, 85, 107).

[Jun22] Jung, Christian and Jörg Dörr: 'Data Usage Control'. *Designing Data Spaces: The Ecosystem Approach to Competitive Advantage*. Ed. by Otto, Boris, Michael ten Hompel, and Stefan Wrobel. Springer, 2022: pp. 129–146 (cit. on p. 71).

[Kha14] Khan, Nawsher, Ibrar Yaqoob, Ibrahim Abaker Targio Hashem, Zakira Inayat, Waleed Kamaleldin Mahmoud Ali, Muhammad Alam, Muhammad Shiraz, and Abdullah Gani: 'Big data: survey, technologies, opportunities, and challenges'. *The scientific world journal* (2014), vol. 2014 (cit. on p. 69).

[Kim99] Kim, Tschangho John: 'Metadata for geo-spatial data sharing: A comparative analysis'. *The Annals of Regional Science* (1999), vol. 33: pp. 171–181 (cit. on p. 8).

[Kir19b] Kirstein, Fabian, Benjamin Dittwald, Simon Dutkowski, Yury Glikman, Sonja Schimmler, and Manfred Hauswirth: 'Linked data in the european data portal: A comprehensive platform for applying dcat-ap'. *Electronic Government: 18th IFIP WG 8.5 International Conference, EGOV 2019, San Benedetto Del Tronto, Italy, September 2–4, 2019, Proceedings 18*. Springer. 2019: pp. 192–204 (cit. on pp. 16, 63, 78, 79).

[Kir19c] Kirstein, Fabian, Simon Dutkowski, Benjamin Dittwald, and Manfred Hauswirth: 'The European Data Portal: Scalable Harvesting and Management of Linked Open Data'. *Proceedings of the ISWC 2019 Satellite Tracks (Posters & Demonstrations, Industry, and Outrageous Ideas) co-located with 18th International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, October 26-30, 2019*. Ed. by Suárez-Figueroa, Mari Carmen, Gong Cheng, Anna Lisa Gentile, Christophe Guéret, C. Maria Keet, and Abraham Bernstein. Vol. 2456. CEUR Workshop Proceedings. CEUR-WS.org, 2019: pp. 321–322 (cit. on pp. 1, 51).

[Kno94] Knoll, Kathleen and Sirkka L. Jarvenpaa: 'Information technology alignment or "fit" in highly turbulent environments: the concept of flexibility'. *Proceedings of the 1994 Computer Personnel Research Conference on Reinventing IS: Managing Information Technology in Changing Organizations, SIGCPR 1994, Alexandria, Virginia, USA, March 24-26, 1994*. Ed. by Ross, Jeanne W. ACM, 1994: pp. 1–14 (cit. on pp. 54, 55).

[Kor19] Korte, Tobias, Martin Fadler, Markus Spiekermann, Christine Legner, and Boris Otto: *Data Catalogs - Integrated Platforms for Matching Data Supply and Demand*. Fraunhofer Verlag, 2019 (cit. on p. 11).

[Kow17] Kowalczyk, Stacy T.: 'Modelling the Research Data Lifecycle'. *Int. J. Digit. Curation* (2017), vol. 12(2): pp. 331–361 (cit. on p. 69).

[Kru16] Kruse, Leona Chandra, Stefan Seidel, and Sandeep Purao: 'Making Use of Design Principles'. *Tackling Society's Grand Challenges with Design Science - 11th International Conference, DESRIST 2016, St. John's, NL, Canada, May 23-25, 2016, Proceedings*. Ed. by Parsons, Jeffrey, Tuure Tuunanen, John Venable, Brian Donnellan, Markus Helfert, and Jim Kenneally. Vol. 9661. Lecture Notes in Computer Science. Springer, 2016: pp. 37–51 (cit. on pp. 4, 5, 105, 110).

[Kuh21] Kuhail, Mohammad Amin, Shahbano Farooq, Rawad Hammad, and Mohammed Bahja: 'Characterizing Visual Programming Approaches for End-User Developers: A Systematic Review'. *IEEE Access* (2021), vol. 9: pp. 14181–14202 (cit. on p. 57).

[Kum21] Kumar, Ganesh, Shuib Basri, Abdullahi Abubakar Imam, Sunder Ali Khowaja, Luiz Fernando Capretz, and Abdullateef Oluwagbemiga Balogun: 'Data harmonization for heterogeneous datasets: a systematic literature review'. *Applied Sciences* (2021), vol. 11(17): p. 8275 (cit. on p. 2).

[Lab20b] Labadie, Clément, Christine Legner, Markus Eurich, and Martin Fadler: 'Fair enough? Enhancing the usage of enterprise data with data catalogs'. *2020 IEEE 22nd Conference on Business Informatics (CBI)*. Vol. 1. IEEE. 2020: pp. 201–210 (cit. on pp. 1–4, 10, 21, 50, 105).

[Lee11] Lee, Jong Seok, Jan Pries-Heje, and Richard L. Baskerville: 'Theorizing in Design Science Research'. *Service-Oriented Perspectives in Design Science Research - 6th International Conference, DESRIST 2011, Milwaukee, WI, USA, May 5-6, 2011. Proceedings*. Ed. by Jain, Hemant K., Atish P. Sinha, and Padmal Vitharana. Vol. 6629. Lecture Notes in Computer Science. Springer, 2011: pp. 1–16 (cit. on pp. 5, 85).

[Lef21] Lefebvre, Hippolyte, Christine Legner, and Martin Fadler: 'Data democratization: toward a deeper understanding'. *Proceedings of the 42nd International Conference on Information Systems, ICIS 2021, Building Sustainability and Resilience with IS: A Call for Action, Austin, TX, USA, December 12-15, 2021*. Ed. by Valacich, Joe S., Anitesh Barua, Ryan T. Wright, Atreyi Kankanhalli, Xitong Li, and Shaila Miranda. Association for Information Systems, 2021 (cit. on pp. 1, 3, 10, 21, 105).

[Leg17] Legner, Christine, Torsten Eymann, Thomas Hess, Christian Matt, Tilo Böhmann, Paul Drews, Alexander Maedche, Nils Urbach, and Frederik Ahlemann: 'Digitalization: Opportunity and Challenge for the Business and Information Systems Engineering Community'. *Bus. Inf. Syst. Eng.* (2017), vol. 59(4): pp. 301–308 (cit. on pp. 1, 105).

[Len14] Lenhardt, W, Stanley Ahalt, Brian Blanton, Laura Christopherson, and Ray Idaszak: 'Data management lifecycle and software lifecycle management in the context of conducting science'. *Journal of Open Research Software* (2014), vol. 2(1) (cit. on p. 69).

[Lin14] Lin, Li, Tingting Liu, Jian Hu, and Jianbiao Zhang: 'A privacy-aware cloud service selection method toward data life-cycle'. *2014 20th IEEE international conference on parallel and distributed systems (ICPADS)*. IEEE. 2014: pp. 752–759 (cit. on p. 69).

[Lur07] Lurie, Nicholas H and Charlotte H Mason: 'Visual representation: Implications for decision making'. *Journal of marketing* (2007), vol. 71(1): pp. 160–177 (cit. on p. 73).

[Ma14] Ma, Xiaogang, Peter Fox, Eric Rozell, Patrick West, and Stephan Zednik: 'Ontology dynamics in a data life cycle: challenges and recommendations from a Geoscience Perspective'. *Journal of Earth Science* (2014), vol. 25: pp. 407–412 (cit. on p. 69).

[Maa10b] Maali, Fadi, Richard Cyganiak, and Vassilios Peristeras: 'Enabling Interoperability of Government Data Catalogues'. *Electronic Government, 9th IFIP WG 8.5 International Conference, EGOV 2010, Lausanne, Switzerland, August 29 - September 2, 2010. Proceedings*. Ed. by Wimmer, Maria A., Jean-Loup Chappelet, Marijn Janssen, and Hans Jochen Scholl. Vol. 6228. Lecture Notes in Computer Science. Springer, 2010: pp. 339–350 (cit. on p. 9).

[Maj18] Majumdar, Debashis, Pradipta Kumar Banerji, and Satyajit Chakrabarti: 'Disruptive technology and disruptive innovation: ignore at your peril!' *Technology Analysis & Strategic Management* (2018), vol. 30(11): pp. 1247–1255 (cit. on p. 54).

[Mar95] March, Salvatore T. and Gerald F. Smith: 'Design and natural science research on information technology'. *Decis. Support Syst.* (1995), vol. 15(4): pp. 251–266 (cit. on p. 23).

[Mer16] Merkel, Angela and Patrizia Mangold: *PODCAST Merkel: Wir müssen uns sputen*. Presse- und Informationsamt der Bundesregierung [Accessed: June 12, 2018]. 2016 (cit. on p. 1).

[Met15] Meth, Hendrik, Benjamin Müller, and Alexander Maedche: 'Designing a Requirement Mining System'. *J. Assoc. Inf. Syst.* (2015), vol. 16(9): p. 2 (cit. on p. 5).

[Mey93] Meyer, Alan D, James B Goes, and Geoffrey R Brooks: 'Organizations reacting to hyperturbulence'. *Organizational change and redesign* (1993), vol.: pp. 66–111 (cit. on p. 54).

[Mia17] Miao, Hui, Ang Li, Larry S. Davis, and Amol Deshpande: 'Towards Unified Data and Lifecycle Management for Deep Learning'. *33rd IEEE International Conference on Data Engineering, ICDE 2017, San Diego, CA, USA, April 19-22, 2017*. IEEE Computer Society, 2017: pp. 571–582 (cit. on p. 69).

[Mic15] Michota, Alexandra and Sokratis Katsikas: 'Designing a seamless privacy policy for social networks'. *Proceedings of the 19th panhellenic conference on informatics*. 2015: pp. 139–143 (cit. on p. 69).

[Moh20] Mohammadi, Nazila Gol, Ludger Goeke, Maritta Heisel, and Mike Surridge: 'Systematic Risk Assessment of Cloud Computing Systems using a Combined Model-based Approach.' *ICEIS (2)*. 2020: pp. 53–66 (cit. on p. 77).

[Möl20] Möller, Frederik, Tobias Moritz Guggenberger, and Boris Otto: 'Towards a method for design principle development in information systems'. *Designing for Digital Transformation. Co-Creating Services with Citizens and Industry: 15th International Conference on Design Science Research in Information Systems and Technology, DESRIST 2020, Kristiansand, Norway, December 2–4, 2020, Proceedings 15*. Springer. 2020: pp. 208–220 (cit. on pp. 25, 26, 105).

[Nel97] Nelson, Kay M. and H. James Nelson: 'Technology Flexibility: Conceptualization, Validation, and Measurement'. *30th Annual Hawaii International Conference on System Sciences (HICSS-30), 7-10 January 1997, Maui, Hawaii, USA*. IEEE Computer Society, 1997: p. 76 (cit. on pp. 54, 55).

[Och09] Ochoa, Xavier and Erik Duval: 'Automatic evaluation of metadata quality in digital repositories'. *Int. J. Digit. Libr.* (2009), vol. 10(2-3): pp. 67–91 (cit. on p. 9).

[Ole23] Olesen-Bagneux, Ole: *The Enterprise Data Catalog - Improve Data Discovery, Ensure Data Governance, and Enable Innovation*. Sebastopol, California: O'Reilly Media, Incorporated, 2023 (cit. on pp. 1, 3, 10, 105).

[Org04] Organization, National Information Standards: *Understanding Metadata*. `https://www.lter.uaf.edu/metadata_files/UnderstandingMetadata.pdf` [Accessed: July 14, 2023]. 2004 (cit. on p. 7).

[Öst10] Österle, Hubert, Robert Winter, and Walter Brenner: *Gestaltungsorientierte Wirtschaftsinformatik: Ein Plädoyer für Rigor und Relevanz*. Infowerk, 2010 (cit. on p. 23).

[Ott15] Otto, Boris: 'Quality and Value of the Data Resource in Large Enterprises'. *Inf. Syst. Manag.* (2015), vol. 32(3): pp. 234–251 (cit. on p. 1).

[Ott11] Otto, Boris and Kristin Weber: 'Data governance'. *Daten-und Informationsqualität: Auf dem Weg zur Information Excellence* (2011), vol.: pp. 277–295 (cit. on p. 2).

[Ott07] Otto, Boris, Kristin Wende, Alexander Schmidt, and Philipp Osl: 'Towards a framework for corporate data quality management'. (2007), vol. (cit. on p. 1).

[Pac18] Pachpor, Nishant N and Prakash S Prasad: 'Improving the performance of system in cloud by using selective deduplication'. *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*. IEEE. 2018: pp. 314–318 (cit. on p. 71).

[Pat16] Patel, Dimple: 'Research data management: a conceptual framework'. *Library review* (2016), vol. 65(4/5): pp. 226–241 (cit. on p. 69).

[Pat19] Patel, Jayesh: 'Bridging data silos using big data integration'. *International Journal of Database Management Systems* (2019), vol. 11(3): pp. 01–06 (cit. on p. 3).

[Pef08] Peffers, Ken, Tuure Tuunanen, Marcus A. Rothenberger, and Samir Chatterjee: 'A Design Science Research Methodology for Information Systems Research'. *J. Manag. Inf. Syst.* (2008), vol. 24(3): pp. 45–77 (cit. on p. 23).

[Pos02] Post, Frits H, Gregory Nielson, and Georges-Pierre Bonneau: 'Data visualization: The state of the art'. (2002), vol. (cit. on p. 73).

[Pru21] Prukalpa: *We Failed to Set Up a Data Catalog 3x. Here's Why.* `https://towardsdatascience.com/our-learnings-from-3-failures-over-5-years-to-set-up-a-data-catalog-fb9778e25d4e` [Accessed: July 14, 2023]. 2021 (cit. on p. 3).

[Qui20] Quimbert, Erwann, Keith G. Jeffery, Claudia Martens, Paul Martin, and Zhiming Zhao: 'Data Cataloguing'. *Towards Interoperable Research Infrastructures for Environmental and Earth Sciences - A Reference Model Guided Approach for Common Challenges*. Ed. by Zhao, Zhiming and Margareta Hellström. Vol. 12003. Lecture Notes in Computer Science. Springer, 2020: pp. 140–161 (cit. on pp. 1, 9, 10, 12, 69, 70, 105).

[Rah12] Rahman, Nayem, Jessica Marz, and Shameem Akhter: 'An ETL Metadata Model for Data Warehousing'. *J. Comput. Inf. Technol.* (2012), vol. 20(2): pp. 95–111 (cit. on p. 8).

[Rea12] Reardon, Joel, Srdjan Capkun, and David A. Basin: 'Data Node Encrypted File System: Efficient Secure Deletion for Flash Memory'. *Proceedings of the 21th USENIX Security Symposium, Bellevue, WA, USA, August 8-10, 2012*. Ed. by Kohno, Tadayoshi. USENIX Association, 2012: pp. 333–348 (cit. on p. 71).

[Rec11] Recker, Jan, Michael Rosemann, Peter F. Green, and Marta Indulska: 'Do Ontological Deficiencies in Modeling Grammars Matter?' *MIS Q.* (2011), vol. 35(1): pp. 57–79 (cit. on pp. 24, 85–87, 106, 110).

[Red98] Redman, Thomas C.: 'the Impact of Poor Data Quality on the Typical Enterprise'. *Commun. ACM* (1998), vol. 41(2): pp. 79–82 (cit. on p. 76).

[Reg15] Regenscheid, Andrew, Larry Feldman, and Gregory Witte: *NIST Special Publication 800-88 Revision 1, Guidelines for Media Sanitization*. Tech. rep. National Institute of Standards and Technology, 2015 (cit. on p. 71).

[Rei09] Reichert, Manfred, Stefanie Rinderle-Ma, and Peter Dadam: 'Flexibility in process-aware information systems'. *Transactions on Petri Nets and Other Models of Concurrency II: Special Issue on Concurrency in Process-Aware Information Systems* (2009), vol.: pp. 115–135 (cit. on p. 54).

[Rez19] Rezig, El Kindi, Lei Cao, Michael Stonebraker, Giovanni Simonini, Wenbo Tao, Samuel Madden, Mourad Ouzzani, Nan Tang, and Ahmed K. Elmagarmid: 'Data Civilizer 2.0: A Holistic Framework for Data Preparation and Analytics'. *Proc. VLDB Endow.* (2019), vol. 12(12): pp. 1954–1957 (cit. on pp. 52, 59).

[Rhy17] Rhyn, Marcel and Ivo Blohm: 'Combining Collective and Artificial Intelligence: towards a Design Theory for Decision Support in Crowdsourcing'. *25th European Conference on Information Systems, ECIS 2017, Guimarães, Portugal, June 5-10, 2017*. Ed. by Ramos, Isabel, Virpi Tuunainen, and Helmut Krcmar. 2017 (cit. on p. 5).

[Ril17] Riley, Jenn: 'Understanding Metadata'. *Washington DC, United States: National Information Standards Organization* (2017), vol. 23. `http://www.niso.org/publications/press/UnderstandingMetadata.pdf` [Accessed: July 14, 2023]: pp. 7–10 (cit. on p. 8).

[Rod16] Rodríguez-Mazahua, Lisbeth, Cristian Aarón Rodríguez-Enríquez, José Luis Sánchez-Cervantes, Jair Cervantes, Jorge Luis García-Alcaraz, and Giner Alor-Hernández: 'A general perspective of Big Data: applications, tools, challenges and trends'. *J. Supercomput.* (2016), vol. 72(8): pp. 3073–3113 (cit. on p. 2).

[Ros10a] Rose, Stuart, Dave Engel, Nick Cramer, and Wendy Cowley: 'Automatic keyword extraction from individual documents'. *Text mining: applications and theory* (2010), vol.: pp. 1–20 (cit. on p. 40).

[Ros10b] Roszkiewicz, Ron: 'Enterprise metadata management: How consolidation simplifies control'. *Journal of Digital Asset Management* (2010), vol. 6: pp. 291–297 (cit. on pp. 3, 7).

[Run20] Runeson, Per, Emelie Engström, and Margaret-Anne D. Storey: 'The Design Science Paradigm as a Frame for Empirical Software Engineering'. *Contemporary Empirical Methods in Software Engineering*. Ed. by Felderer, Michael and Guilherme Horta Travassos. Springer, 2020: pp. 127–147 (cit. on pp. 4, 26).

[Run14] Runeson, Per, Sten Minör, and Johan Svenér: 'Get the cogs in synch: time horizon aspects of industry-academia collaboration'. *WISE'14, Proceedings of the 2014 ACM International Workshop on Long-term Industrial Collaboration on Software Engineering, Vasteras, Sweden, September 16, 2014*. Ed. by Dobrin, Radu, Peter Wallin, Ana C. R. Paiva, and Myra B. Cohen. ACM, 2014: pp. 25–28 (cit. on p. 23).

[Ryd18] Rydning, David Reinsel-John Gantz-John, John Reinsel, and John Gantz: 'The digitization of the world from edge to core'. *Framingham: International Data Corporation* (2018), vol. 16: pp. 1–28 (cit. on p. 2).

[Sab22] Sabot, Francois: 'On the importance of metadata when sharing and opening data'. *BMC Genomic Data* (2022), vol. 23(1): p. 79 (cit. on p. 7).

[Sam22] Samarasinghe, Sasari Sanika Udanjala, Sachithra Lokuge, and Lan Snell: 'Exploring Tenets of Data Democratization'. *26th Pacific Asia Conference on Information Systems, PACIS 2022, Virtual Event / Taipei, Taiwan / Sydney, Australia, July 5-9, 2022*. Ed. by Huang, Ming-Hui, Guy Gable, Christy M. K. Cheung, and Dongming Xu. 2022: p. 336 (cit. on pp. 1, 2, 105).

[San19] Sanchis, Raquel, Óscar García-Perales, Francisco Fraile, and Raul Poler: 'Low-code as enabler of digital transformation in manufacturing industry'. *Applied Sciences* (2019), vol. 10(1): p. 12 (cit. on p. 57).

[Sch18a] Schjerlund, Jonas, Magnus Rotvit Perlt Hansen, and Josefine Gill Jensen: 'Design Principles for Room-Scale Virtual Reality: A Design Experiment in Three Dimensions'. *Designing for a Digital and Globalized World - 13th International Conference, DESRIST 2018, Chennai, India, June 3-6, 2018, Proceedings*. Ed. by Chatterjee, Samir, Kaushik Dutta, and Rangaraja

P. Sundarraj. Vol. 10844. Lecture Notes in Computer Science. Springer, 2018: pp. 3–17 (cit. on p. 109).

[Sch11] Schober, Franz and Judith Gebauer: 'How much to spend on flexibility? Determining the value of information system flexibility'. *Decision Support Systems* (2011), vol. 51(3): pp. 638–647 (cit. on p. 55).

[Sei11] Sein, Maung K, Ola Henfridsson, Sandeep Purao, Matti Rossi, and Rikard Lindgren: 'Action design research'. *MIS quarterly* (2011), vol.: pp. 37–56 (cit. on pp. 4, 23).

[Sen04b] Sen, Arun: 'Metadata management: past, present and future'. *Decis. Support Syst.* (2004), vol. 37(1): pp. 151–173 (cit. on pp. 7, 10, 11).

[Sha16a] Shameli-Sendi, Alireza, Rouzbeh Aghababaei-Barzegar, and Mohamed Cheriet: 'Taxonomy of information security risk assessment (ISRA)'. *Computers & security* (2016), vol. 57: pp. 14–30 (cit. on p. 77).

[Sha16c] Shanmugam, Srinivasan and Gokul Seshadri: 'Aspects of data cataloguing for enterprise data platforms'. *2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS)*. IEEE. 2016: pp. 134–139 (cit. on pp. 1, 3, 4, 10, 66, 75, 78, 105).

[She17] Shestakofsky, Benjamin: 'Working algorithms: Software automation and the future of work'. *Work and Occupations* (2017), vol. 44(4): pp. 376–423 (cit. on p. 50).

[Spi18] Spiekermann, Markus, Daniel Tebernum, Sven Wenzel, and Boris Otto: 'A metadata model for data goods'. *Proceedings of the Multikonferenz Wirtschaftsinformatik (MKWI) 2018*. Ed. by Drews, Paul, Burkhardt Funk, Peter Niemeyer, and Lin Xie. Vol. 2018. ISBN 978-3-935786-72-0. https://www.leuphana.de/fileadmin/user_upload/Forschungseinrichtungen/iis/files/MKWI2018/MKWI2018_Band1.pdf [Accessed: August 16, 2022]. 2018: pp. 326–337 (cit. on pp. 30, 31, 47, 63, 67, 106).

[Sta16] Stadler, Jennifer G, Kipp Donlon, Jordan D Siewert, Tessa Franken, and Nathaniel E Lewis: 'Improving the efficiency and ease of healthcare analysis through use of data visualization dashboards'. *Big data* (2016), vol. 4(2): pp. 129–135 (cit. on p. 74).

[Ste10] Steele, Julie and Noah Iliinsky: *Beautiful visualization: Looking at data through the eyes of experts*. " O'Reilly Media, Inc.", 2010 (cit. on p. 73).

[Sto15] Stojnic, Nenad: 'Self-organizing distributed workflow management'. PhD thesis. University_of_Basel, 2015 (cit. on pp. 52, 53).

[Str24] Stritesky, Joseph, Natalie Hallak, and Danrlei Alves: *What are the signs of an existing Data Catalog failing*. https://atlan.com/signs-of-a-traditional-data-catalog-failing/ [Accessed: June 04, 2024]. Atlan. 2024 (cit. on p. 3).

[Sum24] Sumaria, Rupal: *Case Study Penguin Random House*. https://data.world/case-studies/penguin-random-house/ [Accessed: May 17, 2024]. 2024 (cit. on p. 1).

[Szu23] Szukits, Ágnes and Péter Móricz: 'Towards data-driven decision making: the role of analytical culture and centralization efforts'. *Review of Managerial Science* (2023), vol.: pp. 1–39 (cit. on pp. 1, 105).

[Tal14] Tallon, Paul P., Ronald V. Ramirez, and James E. Short: 'The Information Artifact in IT Governance: Toward a Theory of Information Governance'. *J. Manag. Inf. Syst.* (2014), vol. 30(3): pp. 141–178 (cit. on p. 1).

[Tam07] Tambouris, Efthimios, Nikos Manouselis, and Constantina I. Costopoulou: 'Metadata for digital collections of e-government resources'. *Electron. Libr.* (2007), vol. 25(2): pp. 176–192 (cit. on p. 9).

[Teb23a] Tebernum, Daniel, Marcel Altendeitering, and Falk Howar: 'A Survey-Based Evaluation of the Data Engineering Maturity in Practice'. *Data Management Technologies and Applications*. Ed. by Cuzzocrea, Alfredo, Oleg Gusikhin, Slimane Hammoudi, and Christoph Quix. DOI https://doi.org/10.1007/978-3-031-37890-4_1. https://link.springer.com/chapter/10.1007/978-3-031-37890-4_1 [Accessed: August 2, 2023]. Cham: Springer Nature Switzerland, 2023: pp. 1–23 (cit. on pp. 45, 47, 71, 76, 107).

[Teb21] Tebernum, Daniel, Marcel Altendeitering, and Falk Howar: 'DERM: A Reference Model for Data Engineering'. *Proceedings of the 10th International Conference on Data Science, Technology and Applications, DATA 2021, Online Streaming, July 6-8, 2021*. Ed. by Quix, Christoph, Slimane Hammoudi, and Wil M. P. van der Aalst. DOI https://doi.org/10.5220/0010517301650175. https://www.scitepress.org/PublishedPapers/2021/105173/105173.pdf [Accessed: August 16, 2022]. SCITEPRESS, 2021: pp. 165–175 (cit. on pp. 45, 47, 71, 107).

[Teb23b] Tebernum, Daniel and Sergej Atamantschuk: *DIVA - Data Inventory and Valuation Approach*. Version 4.1.0. https://github.com/FraunhoferISST/diva [Accessed: June 30, 2023]. June 2023 (cit. on p. 65).

[Teb20] Tebernum, Daniel and Dustin Chabrowski: 'A Conceptual Framework for a Flexible Data Analytics Network'. *Proceedings of the 9th International Conference on Data Science, Technology and Applications, DATA 2020, Lieusaint, Paris, France, July 7-9, 2020*. Ed. by Hammoudi, Slimane, Christoph Quix, and Jorge Bernardino. DOI https://doi.org/10.5220/0009827402330233. https://www.scitepress.org/PublishedPapers/2020/98274/98274.pdf [Accessed: August 16, 2022]. SciTePress, 2020: pp. 223–233 (cit. on pp. 38, 39, 47, 53, 60, 65, 106).

[Teb23c] Tebernum, Daniel and Falk Howar: 'Structuring the End of the Data Life Cycle'. *Proceedings of the 12th International Conference on Data Science, Technology and Applications - DATA*. DOI https://doi.org/10.5220/0011999300003541. https://www.scitepress.org/Papers/2023/119993/119993.pdf [Accessed: August 2, 2023]. INSTICC. SciTePress, 2023: pp. 207–218 (cit. on pp. 45, 47, 71, 107).

[Teb23d] Tebernum, Daniel and Falk Howar: 'Treating the End of the Data Life Cycle as a First-Class Citizen in Data Engineering'. *Wirtschaftsinformatik 2023 Proceedings. 8*. https://aisel.aisnet.org/wi2023/8. https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1007&context=wi2023 [Accessed: Oktober 4, 2023]. 2023 (cit. on pp. 45–47, 62, 63, 70–72, 107).

[Teb18] Tebernum, Daniel, Markus Spiekermann, Sven Wenzel, and Boris Otto: 'Risikobewertungen in Datennetzwerken'. *D · A · CH Security 2018 : Bestandsaufnahme, Konzepte, Anwendungen, Perspektiven*. Ed. by Schartner, Peter and Norbert Pohlmann. ISBN 978-3-00-060424-9. https://www.syssec.at/de/veranstaltungen/dachsecurity2018/papers/DACH_Security_2018_Paper_23A1.pdf [Accessed: August 16, 2022]. Frechen : syssec, 2018: pp. 287–297 (cit. on pp. 33, 47, 53, 58, 77, 80, 106).

[Teg99] Tegarden, David P.: 'Business Information Visualization'. *Commun. Assoc. Inf. Syst.* (1999), vol. 1: p. 4 (cit. on p. 73).

[Tel14] Telea, Alexandru C: *Data visualization: principles and practice*. CRC Press, 2014 (cit. on p. 73).

[Tra18] Trajanov, Dimitar, Vladimir Zdraveski, Riste Stojanov, and Ljupco Kocarev: 'Dark data in internet of things (IOT): challenges and opportunities'. *7th Small Systems Simulation Symposium*. 2018: pp. 1–8 (cit. on p. 71).

[Tuu18] Tuunanen, Tuure and Ken Peffers: 'Population targeted requirements acquisition'. *Eur. J. Inf. Syst.* (2018), vol. 27(6): pp. 686–711 (cit. on p. 109).

[Vai04] Vaishnavi, V and W Kuechler: *Design Science Research in Information Systems*. (updated in 2017 and 2019 by Vaishnavi, V. and Stacey, P.); last updated November 24, 2021. 2004. URL: http://www.desrist.org/design-research-in-information-systems/ (cit. on pp. 4, 23, 29).

[Ven17] Venable, John R., Jan Pries-Heje, and Richard L. Baskerville: 'Choosing a Design Science Research Methodology'. *Australasian Conference on Information Systems, ACIS 2017, Hobart, Tasmania, Australia, 4-6 December 2017*. 2017: p. 112 (cit. on p. 23).

[Ven16] Venable, John R., Jan Pries-Heje, and Richard L. Baskerville: 'FEDS: a Framework for Evaluation in Design Science Research'. *Eur. J. Inf. Syst.* (2016), vol. 25(1): pp. 77–89 (cit. on pp. 24, 85, 110).

[Ver13] Verbert, Katrien, Erik Duval, Joris Klerkx, Sten Govaerts, and José Luis Santos: 'Learning analytics dashboard applications'. *American Behavioral Scientist* (2013), vol. 57(10): pp. 1500–1509 (cit. on p. 74).

[Vos19] Vos, Marina De, Sabrina Kirrane, Julian A. Padget, and Ken Satoh: 'ODRL Policy Modelling and Compliance Checking'. *Rules and Reasoning - Third International Joint Conference, RuleML+RR 2019, Bolzano, Italy, September 16-19, 2019, Proceedings*. Ed. by Fodor, Paul, Marco Montali, Diego Calvanese, and Dumitru Roman. Vol. 11784. Lecture Notes in Computer Science. Springer, 2019: pp. 36–51 (cit. on p. 63).

[Wac22] Wache, Hendrik, Frederik Möller, Thorsten Schoormann, Gero Strobel, and Dimitri Petrik: 'Exploring the abstraction levels of design principles: the case of chatbots'. (2022), vol. (cit. on pp. 26, 27).

[Wag17] Wagenknecht, Thomas, René Filpe, and Christof Weinhardt: 'Towards a design theory of computer-supported organizational participation'. *J. Enterp. Inf. Manag.* (2017), vol. 30(1): pp. 188–202 (cit. on p. 109).

[Wan96] Wang, Richard Y and Diane M Strong: 'Beyond accuracy: What data quality means to data consumers'. *Journal of management information systems* (1996), vol. 12(4): pp. 5–33 (cit. on pp. 76, 78).

[Wan09] Wang, Wenguang, Andreas Tolk, and Weiping Wang: 'The levels of conceptual interoperability model: applying systems engineering principles to M&S'. *Proceedings of the 2009 Spring Simulation Multiconference, SpringSim 2009, San Diego, California, USA, March 22-27, 2009*. Ed. by Wainer, Gabriel A., Clifford A. Shaffer, Robert M. McGraw, and Michael J. Chinni. SCS/ACM, 2009 (cit. on pp. 61, 64).

[Wan17b] Wangen, Gaute: 'Information security risk assessment: a method comparison'. *Computer* (2017), vol. 50(4): pp. 52–61 (cit. on p. 77).

[Waw06] Wawrzyniak, Dariusz: 'Information security risk assessment model for risk management'. *Trust and Privacy in Digital Business: Third International Conference, TrustBus 2006, Kraków, Poland, September 4-8, 2006. Proceedings 3*. Springer. 2006: pp. 21–30 (cit. on p. 77).

[Web02] Webster, Jane and Richard T. Watson: 'Analyzing the Past to Prepare for the Future: Writing a Literature Review'. *MIS Q.* (2002), vol. 26(2) (cit. on pp. 19, 23).

[Weg96] Wegner, Peter: 'Interoperability'. *ACM Comput. Surv.* (1996), vol. 28(1): pp. 285–287 (cit. on p. 60).

[Wei98] Weibel, Stuart, John A. Kunze, Carl Lagoze, and Misha Wolf: 'Dublin Core Metadata for Resource Discovery'. *RFC* (1998), vol. 2413: pp. 1–8 (cit. on p. 8).

[Wie14] Wieringa, Roel J.: *Design Science Methodology for Information Systems and Software Engineering*. Springer, 2014 (cit. on p. 23).

[Wil14] Wilbanks, Bryan A and Patsy A Langford: 'A review of dashboards for data analytics in nursing'. *CIN: Computers, Informatics, Nursing* (2014), vol. 32(11): pp. 545–549 (cit. on p. 74).

[Wil16] Wilkinson, Mark D, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al.: 'The FAIR Guiding Principles for scientific data management and stewardship'. *Scientific data* (2016), vol. 3(1): pp. 1–9 (cit. on pp. 2, 3).

[Wis16] Wissik, Tanja and Matej Ďurčo: 'Research data workflows: from research data lifecycle models to institutional solutions'. *Selected papers from the CLARIN annual conference 2015, October 14–16, 2015, Wroclaw, Poland*. 123. Linköping University Electronic Press. 2016: pp. 94–107 (cit. on p. 69).

[Woh21] Wohlin, Claes and Per Runeson: 'Guiding the selection of research methodology in industry-academia collaboration in software engineering'. *Inf. Softw. Technol.* (2021), vol. 140: p. 106678 (cit. on p. 23).

[Woo97] Woodruff, Allison and Michael Stonebraker: 'Supporting Fine-grained Data Lineage in a Database Visualization Environment'. *Proceedings of the Thirteenth International Conference on Data Engineering, April 7-11, 1997, Birmingham, UK*. Ed. by Gray, W. A. and Per-Åke Larson. IEEE Computer Society, 1997: pp. 91–102 (cit. on p. 75).

[Yi07] Yi, Ji Soo, Youn ah Kang, John T. Stasko, and Julie A. Jacko: 'Toward a Deeper Understanding of the Role of Interaction in Information Visualization'. *IEEE Trans. Vis. Comput. Graph.* (2007), vol. 13(6): pp. 1224–1231 (cit. on p. 74).

[Zai17] Zaidi, Ehtisham, Guido De Simoni, Roxane Edjlali, and Alan D Duncan: 'Data catalogs are the new black in data management and analytics'. *Gartner, Consultancy Report* (2017), vol. (cit. on pp. 3, 9–12).

[Zen18] Zeng, Jing and Keith W Glaister: 'Value creation from big data: Looking inside the black box'. *Strategic Organization* (2018), vol. 16(2): pp. 105–140 (cit. on p. 2).

Data Catalog Corpus

[Ade17] Adel Rezk, M., A. Ojo, and I.A. Hassan: 'Mining governmental collaboration through semantic profiling of open data catalogues and publishers'. English. *IFIP Advances in Information and Communication Technology* (2017), vol. 506. Ed. by Camarinha-Matos, L.M., R. Fornasiero, and H. Afsarmanesh. cited By 1: pp. 253–264 (cit. on pp. 1, 21, 49, 50, 60, 62, 66, 67).

[Aik20] Aikoh, K., Y. Isoda, and K. Sugimoto: 'Data profiling method for metadata management'. English. Ed. by Webb, G., Z. Zhang, V.S. Tseng, G. Williams, M. Vlachos, and L. Cao. cited By 0. Institute of Electrical and Electronics Engineers Inc., 2020: pp. 779–780 (cit. on pp. 20, 49, 50).

[Alv22] Alvarez, R., C. González-Mora, I. Garrigós, et al.: 'A Metadata-Driven Tool for FAIR Data Production in Citizen Science Platforms'. English. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2022), vol. 13362 LNCS. Ed. by Di Noia, T., I.-Y. Ko, M. Schedl, and C. Ardito. cited By 0: pp. 465–468 (cit. on pp. 60, 61).

[Ate14] Atemezing, G.A., N. Abadie, R. Troncy, et al.: 'Publishing reference Geodata on the web: Opportunities and challenges for IGN France'. English. Ed. by Taylor, K., R. Gruetter, K. Janowicz, M. Compton, D. Kolas, K. Kyzirakos, and M. Perry. Vol. 1401. cited By 2. CEUR-WS, 2014: pp. 9–20 (cit. on pp. 60, 62).

[Att15] Attard, J., F. Orlandi, S. Scerri, et al.: 'A systematic review of open government data initiatives'. English. *Government Information Quarterly* (2015), vol. 32(4). cited By 500: pp. 399–418 (cit. on pp. 10, 14, 77).

[Ayf21] Ayfantopoulou, G., J. Burrieza Galán, A. Masegosa, et al.: 'Cataloging and Assessing City-scale Mobility Data'. English. *Advances in Intelligent Systems and Computing* (2021), vol. 1278. Ed. by Nathanail E.G. Adamos G., Karakikes I. cited By 1: pp. 1139–1148 (cit. on p. 77).

[Bar22b] Barthelemy, F., M. Cochez, I. Dimitriadis, et al.: 'Towards a standard-based open data ecosystem: analysis of DCAT-AP use at national and European level'. English. *Electronic Government* (2022), vol. 18(2). cited By 1: pp. 137–180 (cit. on pp. 60, 62).

[Bas18b] Bastião Silva, L., A. Trifan, and J. Luís Oliveira: 'MONTRA: An agile architecture for data publishing and discovery'. English. *Computer Methods and Programs in Biomedicine* (2018), vol. 160. cited By 19: pp. 33–42 (cit. on pp. 54, 55).

[Ben14] Benedetti, F., S. Bergamaschi, and L. Po: 'A visual summary for linked open data sources'. English. Ed. by Horridge, M., M. Rospocher, and J. van Ossenbruggen. Vol. 1272. cited By 27. CEUR-WS, 2014: pp. 173–176 (cit. on pp. 20, 73).

[Ber22] Bernhauer, D., M. Nečaský, P. Škoda, et al.: 'Open dataset discovery using context-enhanced similarity search'. English. *Knowledge and Information Systems* (2022), vol. 64(12). cited By 0: pp. 3265–3291 (cit. on pp. 20, 66, 67).

[Bog21] Bogdanović, M., N. Veljković, M. Frtunić Gligorijević, et al.: 'On revealing shared conceptualization among open datasets'. English. *Journal of Web Semantics* (2021), vol. 66. cited By 2 (cit. on pp. 60, 62).

[Bor22] Borges, V., N.Q. de Oliveira, and M.L. Machado Campos: 'A Multi-level Ontology-based Approach for Descriptors of Catalogued Resources'. English. Ed. by Parente de Oliveira, J.M., L.F. Garcia, E.R. Felipe, and D. Schmidt. Vol. 3346. cited By 0. CEUR-WS, 2022: pp. 46–59 (cit. on pp. 1, 10, 21, 60, 61, 105).

[Bug22] Bugbee, K., R. Ramachandran, A. Acharya, et al.: 'Selecting Approaches for Enabling Enterprise Data Search: NASA's Science Mission Directorate (SMD) Catalog'. English. Vol. 2022-July. cited By 1. Institute of Electrical and Electronics Engineers Inc., 2022: pp. 6836–6839 (cit. on pp. 1, 10, 14, 15, 20, 68).

[Cap17] Cappello, P., M. Comerio, and I. Celino: 'Botdcat-ap: An extension of the DCAT application Profile for describing datasets for chatbot systems'. English. Ed. by Demidova, E., J. Breslin, S. Dietze, and J. Szymanski. Vol. 1927. cited By 0. CEUR-WS, 2017 (cit. on pp. 20, 60, 61, 63).

[Car11] Careem, M. and D. Karunarathne: 'Enhancing data integration and discovery with ad hoc registries'. English. cited By 0. 2011: pp. 439–442 (cit. on pp. 49, 50).

[Car15] Carvalho, P., P. Hitzelberger, B. Otjacques, et al.: 'Using information visualization to support open data integration'. English. *Communications in Computer and Information Science* (2015), vol. 178. cited By 2 (cit. on pp. 20, 21, 73).

[Cha20] Chapman, A., E. Simperl, L. Koesten, et al.: 'Dataset search: a survey'. English. *VLDB Journal* (2020), vol. 29(1). cited By 92: pp. 251–272.

[Che22] Cherradi, M., A. EL Haddadi, and H. Routaib: 'Data Lake Management Based on DLDS Approach'. English. *Smart Innovation, Systems and Technologies* (2022), vol. 237. Ed. by Ben Ahmed, M., H.L. Teodorescu, T. Mazri, P. Subashini, and A. Boudhir. cited By 2: pp. 679–690 (cit. on pp. 1, 3, 10, 105).

[Cho20] Choi, M.-Y., C.-J. Moon, and S.-J. Jung: 'Building methods of intelligent data catalog based on graph database for data sharing platform'. English. *ICIC Express Letters, Part B: Applications* (2020), vol. 11(10). cited By 0: pp. 953–959 (cit. on pp. 66–68).

[Cho22] Chokki, A.P., C. Alexopoulos, S. Saxena, et al.: 'Metadata quality matters in open government data (OGD) evaluation! An empirical investigation of OGD portals of the GCC constituents'. English. *Transforming Government: People, Process and Policy* (2022), vol. cited By 0 (cit. on p. 77).

[Cla17] Clarke, S.S., J. Davies, and M. Fisher: 'The internet of things – technical challenges for interoperability'. English. *Communications in Computer and Information Science* (2017), vol. 783. Ed. by Mayr, H.C., V. Ermolayev, M. Nikitchenko, A. Ginige, D. Plexousakis, A. Spivakovskiy, and G. Zholtkevych. cited By 0: pp. 3–13 (cit. on pp. 60, 62).

[Cor10] Corbel, S. and T. Poulet: 'WAGCoE data catalogue for geothermal exploration'. English. cited By 1. 2010: pp. 89–94 (cit. on p. 20).

[Cyg10] Cyganiak, R., F. Maali, and V. Peristeras: 'Self-service linked government data with dcat and gridworks'. English. cited By 15. 2010 (cit. on pp. 60, 62).

[Cza17] Czajkowski, K., C. Kesselman, and R. Schuler: 'ERMRest: A collaborative data catalog with fine grain access control'. English. cited By 1. Institute of Electrical and Electronics Engineers Inc., 2017: pp. 510–517 (cit. on pp. 54, 55).

[Dag16] Daga, E., M. D'Aquin, A. Adamou, et al.: 'Addressing exploitability of Smart City data'. English. cited By 14. Institute of Electrical and Electronics Engineers Inc., 2016 (cit. on pp. 69, 70).

[Di16] Di, W. and Z. Jun: 'Release of metadata of open data of Chinese local governmen base on Drupal'. English. cited By 0. Computers and Industrial Engineering, 2016 (cit. on pp. 60, 62).

[Dib20] Dibowski, H., S. Schmid, Y. Svetashova, et al.: 'Using semantic technologies to manage a data lake: Data catalog, provenance and access control'. English. Ed. by Liebig T. Fokoue A., Wu Z. Vol. 2757. cited By 3. CEUR-WS, 2020: pp. 65–80.

[Dih15] Dihé, P., R. Denzer, and S. Schlobinski: 'An information model for a water information platform'. English. *IFIP Advances in Information and Communication Technology* (2015), vol. 448. Ed. by Denzer, R., R.M. Argent, G. Schimak, and J. Hrebicek. cited By 1: pp. 91–101 (cit. on pp. 20, 60, 62).

[Dor21] Dorobat, I.C. and V. Posea: 'Open Data Indicator: An Accumulative Methodology for Measuring the Quality of Open Government Data'. English. cited By 0. Institute of Electrical and Electronics Engineers Inc., 2021 (cit. on pp. 20, 77).

[Dos19] Dos Reis C.P., Jr., W.M.C. Da Silva, L.C.B. Martins, et al.: 'Enhancing open government data with data provenance'. English. cited By 0. Association for Computing Machinery, Inc, 2019: pp. 142–149 (cit. on pp. 66, 67).

[Ehr21a] Ehrlinger, L., J. Schrott, M. Melichar, et al.: 'Data Catalogs: A Systematic Literature Review and Guidelines to Implementation'. English. *Communications in Computer and Information Science* (2021), vol. 1479 CCIS. Ed. by Kotsis, G., A.M. Tjoa, I. Khalil, B. Moser, A. Mashkoor, J. Sametinger, A. Fensel, J. Martinez-Gil, L. Fischer, G. Czech, F. Sobieczky, and S. Khan. cited By 0: pp. 148–158 (cit. on pp. 1, 21, 60, 61, 66, 67).

[Eic22a] Eichler, R., C. Gröger, E. Hoos, et al.: 'Data Shopping — How an Enterprise Data Marketplace Supports Data Democratization in Companies'. English. *Lecture Notes in Business Information Processing* (2022), vol. 452. cited By 3: pp. 19–26 (cit. on pp. 1, 3, 10, 14, 21, 105).

[Eic22b] Eichler, R., C. Gröger, E. Hoos, et al.: 'From Data Asset to Data Product – The Role of the Data Provider in the Enterprise Data Marketplace'. English. *Communications in Computer and Information Science* (2022), vol. 1603 CCIS. Ed. by Barzen J. Leymann F., Dustdar S. cited By 1: pp. 119–138 (cit. on pp. 1, 10).

[Fen17] Feng, J., S. Kong, B. Du, et al.: 'Research on Faceted Search Method for Water Data Catalogue Service'. English. Ed. by Qiu, M. cited By 1. Institute of Electrical and Electronics Engineers Inc., 2017: pp. 70–75 (cit. on p. 20).

[Fer20] Fernandez, R.C., P. Subramaniam, and M.J. Franklin: 'Data market platforms: Trading data assets to solve data problems'. English. *Proceedings of the VLDB Endowment* (2020), vol. 13(11). cited By 36: pp. 1933–1947 (cit. on p. 14).

[Fiz20] Fize, J., M. Roche, and M. Teisseire: 'Could spatial features help the matching of textual data?' English. *Intelligent Data Analysis* (2020), vol. 24(5). cited By 2: pp. 1043–1064 (cit. on pp. 49, 50).

[Fra05a] Franklin, M., A. Halevy, and D. Maier: 'From databases to dataspaces: A new abstraction for information management'. English. *SIGMOD Record* (2005), vol. 34(4). cited By 530: pp. 27–33.

[Fri14] Friddell, J.E., E.F. LeDrew, and W.F. Vincent: 'The polar data catalogue: Best practices for sharing and archiving Canada's polar data'. English. *Data Science Journal* (2014), vol. 13. cited By 10: PDA1–PDA7 (cit. on pp. 20, 61).

[Frt21] Frtunic Gligorijevic, M., M. Bogdanovic, N. Veljkovic, et al.: 'Open data categorization based on formal concept analysis'. English. *IEEE Transactions on Emerging Topics in Computing* (2021), vol. 9(2). cited By 6: pp. 571–581 (cit. on pp. 49, 50).

[Gon07] Gonçalves, M.A., B.L. Moreira, E.A. Fox, et al.: '"What is a good digital library?" - A quality model for digital libraries'. English. *Information Processing and Management* (2007), vol. 43(5). cited By 97: pp. 1416–1437 (cit. on pp. 20, 77).

[Gre22] Green, A. and K. Faith Lawrence: 'The Shock of the New: Testing the Pan-Archival Linked Data Catalogue with Users'. English. Ed. by Candela L., Silvello G. Vol. 3246. cited By 0. CEUR-WS, 2022: pp. 108–115.

[Hal16] Halevy, A., F. Korn, N.F. Noy, et al.: 'Goods: Organizing Google's datasets'. English. Vol. 26-June-2016. cited By 125. Association for Computing Machinery, 2016: pp. 795–806 (cit. on pp. 20, 49, 50, 66, 67).

[Han13] Hanafusa, Y., H. Saito, and Y. Abe: 'The jamstec metadata publication and search system'. English. *Data Science Journal* (2013), vol. 12. cited By 1: WDS221–WDS224 (cit. on p. 20).

[Has20] Hassine, S.B. and D. Clément: 'Open Data Quality Dimensions and Metrics: State of the Art and Applied Use Cases'. English. *Lecture Notes in Business Information Processing* (2020), vol. 394. Ed. by Abramowicz, W. and G. Klein. cited By 2: pp. 311–323 (cit. on pp. 20, 77).

[Hey15] Heyvaert, P., P. Colpaert, R. Verborgh, et al.: 'Merging and enriching DCAT feeds to improve discoverability of datasets'. English. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2015), vol. 9341. Ed. by Zimmermann, A., S. Villata, J. Breslin, C. Faron-Zucker, F. Gandon, and C. Gueret. cited By 3: pp. 67–71 (cit. on pp. 60, 61).

[Hod21] Hodson, J. and A. Spezzatti: 'Hidden in Plain Sight: Building a Global Sustainable Development Data Catalogue'. English. *Lecture Notes in Networks and Systems* (2021), vol. 154. Ed. by Fong S. Dey N., Joshi A. cited By 2: pp. 803–811 (cit. on pp. 49, 50).

[Hol19] Holl, P. and K. Gossling: 'Midas: Towards an Interactive Data Catalog'. English. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2019), vol. 11721 LNCS. Ed. by Gadepally, V., T. Mattson, Stonebrakerm M., F. Wang, G. Luo, Y. Laing, and A. Dubovitskaya. cited By 0: pp. 128–138 (cit. on p. 73).

[Jef14] Jeffery, K.G. and A. Asserson: 'Data intensive science: Shades of grey'. English. Vol. 33. cited By 4. Elsevier B.V., 2014: pp. 223–230 (cit. on pp. 60, 62).

[Jia19] Jiang, S., T.F. Hagelien, M. Natvig, et al.: 'Ontology-Based Semantic Search for Open Government Data'. English. cited By 13. Institute of Electrical and Electronics Engineers Inc., 2019: pp. 7–15 (cit. on pp. 20, 66, 67).

[Joh90] Johnson, Margaret A. and Margaret Cribbs: 'Data catalogs for archive systems'. English. cited By 0. Publ by IEEE, Piscataway, NJ, United States, 1990: pp. 98–101 (cit. on pp. 10, 20, 60, 61).

[Ker15] Kern, D. and B. Mathiak: 'Are there any differences in data set retrieval compared to well-known literature retrieval?' English. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2015), vol. 9316. Ed. by Mazurek, C., M. Werla, and S. Kapidakis. cited By 29: pp. 197–208 (cit. on p. 77).

[Kim21] Kim, D., M.-S. Gil, M.C. Nguyen, et al.: 'Comprehensive Knowledge Archive Network harvester improvement for efficient open-data collection and management'. English. *ETRI Journal* (2021), vol. 43(5). cited By 4: pp. 835–855 (cit. on pp. 49, 50, 69, 70).

[Kir23] Kirstein, F., A. Altenbernd, S. Schimmler, et al.: 'A Decentralised Persistent Identification Layer for DCAT Datasets'. English. cited By 0. Association for Computing Machinery, Inc, 2023: pp. 1424–1427 (cit. on pp. 20, 60, 61).

[Kir19a] Kirstein, F., B. Dittwald, S. Dutkowski, et al.: 'Linked Data in the European Data Portal: A Comprehensive Platform for Applying DCAT-AP'. English. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2019), vol. 11685 LNCS. Ed. by Lindgren, I., M. Janssen, H. Lee, A. Polini, M.P. Rodriguez Bolivar, H.J. Scholl, and E. Tambouris. cited By 15: pp. 192–204 (cit. on pp. 20, 49, 50, 60, 61, 77).

[Klí19] Klímek, J.: 'Reflections on: DCAT-AP representation of Czech national open data catalog and its impact'. English. Ed. by d'Amato, C. and L. Kagal. Vol. 2576. cited By 0. CEUR-WS, 2019 (cit. on p. 77).

[Klí18] Klímek, J. and P. Škoda: 'Linkedpipes DCAT-AP viewer: A native DCAT-AP data catalog'. English. Ed. by Erp, M. van, K. Srinivas, C. Fortuna, M. Atre, and Lopez. V. Vol. 2180. cited By 1. CEUR-WS, 2018 (cit. on pp. 60, 61).

[Kop21] Kopsachilis, V. and M. Vaitis: 'Geolod: A spatial linked data catalog and recommender'. English. *Big Data and Cognitive Computing* (2021), vol. 5(2). cited By 4 (cit. on p. 73).

[Kře19] Křemen, P. and M. Nečaský: 'Improving discoverability of open government data with rich metadata descriptions using semantic government vocabulary'. English. *Journal of Web Semantics* (2019), vol. 55. cited By 13: pp. 1–20 (cit. on pp. 1, 20, 60, 62).

[Kub18] Kubler, S., J. Robert, S. Neumaier, et al.: 'Comparison of metadata quality in open data portals using the Analytic Hierarchy Process'. English. *Government Information Quarterly* (2018), vol. 35(1). cited By 77: pp. 13–29 (cit. on pp. 20, 77).

[Kub16] Kubler, S., J. Robert, Y.L. Traon, et al.: 'Open data portal quality comparison using AHP'. English. Ed. by Kim, Y. and S.M. Liu. Vol. 08-10-June-2016. cited By 21. Association for Computing Machinery, 2016: pp. 397–407 (cit. on pp. 20, 77).

[Kuč13] Kučera, J., D. Chlapek, and M. Nečaský: 'Open government data catalogs: Current approaches and quality perspective'. English. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2013), vol. 8061 LNCS. cited By 52: pp. 152–166 (cit. on pp. 10, 20, 77).

[Lab20a] Labadie, C., C. Legner, M. Eurich, et al.: 'FAIR Enough? Enhancing the Usage of Enterprise Data with Data Catalogs'. English. Ed. by Aier, S., J. Gordijn, H.A. Proper, and J. Verelst. Vol. 1. cited By 8. Institute of Electrical and Electronics Engineers Inc., 2020: pp. 201–210 (cit. on pp. 1, 10, 21, 49, 54, 55, 66, 67, 77).

[Lac17] Lacasta, J., F.J. Lopez-Pellicer, B. Espejo-García, et al.: 'Aggregation-based information retrieval system for geospatial data catalogs'. English. *International Journal of Geographical Information Science* (2017), vol. 31(8). cited By 7: pp. 1583–1605 (cit. on pp. 66, 67).

[Lee12] Lee, H.J. and M. Sohn: 'Construction of tag-based dynamic data catalog (TaDDCat) using ontology'. English. cited By 1. 2012: pp. 697–702 (cit. on pp. 20, 66, 67).

[Lee16] Lee, M., E. Almirall, and J. Wareham: 'Open data and civic apps: First-generation failures, second- generation improvements'. English. *Communications of the ACM* (2016), vol. 59(1). cited By 75: pp. 82–89 (cit. on pp. 60–62).

[Li22] Li, A.W., L.S. Sinnamon, and R. Kopak: 'Exploring learning opportunities for students in open data portal use across data literacy levels'. English. *Information and Learning Science* (2022), vol. 123(9-10). cited By 0: pp. 601–620.

[Maa10a] Maali, F., R. Cyganiak, and V. Peristeras: 'Enabling interoperability of government data catalogues'. English. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2010), vol. 6228 LNCS. cited By 48: pp. 339–350 (cit. on pp. 10, 20, 61).

[Mar13] Marienfeld, F., I. Schieferdecker, E. Lapi, et al.: 'Metadata aggregation at govData.de - An experience report'. English. cited By 11. 2013 (cit. on pp. 10, 49, 50, 61).

[Mar11] Martin, M., M. Kaltenböck, H. Nagy, et al.: 'The open government data stakeholder survey'. English. Vol. 739. cited By 1. 2011 (cit. on p. 10).

[Mil15] Milić, P., N. Veljković, and L. Stoimenov: 'Linked relations architecture for production and consumption of linksets in open government data'. English. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2015), vol. 9373. Ed. by Lamersdorf, W., J. Hidders, M. Janssen, B. Klievink, A. Zuiderwijk, M. Mantymaki, and B. van Loenen. cited By 5: pp. 212–222 (cit. on pp. 66, 67).

[Neu19] Neumaier, S. and A. Polleres: 'Enabling Spatio-Temporal Search in Open Data'. English. *Journal of Web Semantics* (2019), vol. 55. cited By 14: pp. 21–36.

[Neu18a] Neumaier, S., V. Savenkov, and A. Polleres: 'Geo-semantic labelling of open data'. English. Vol. 137. cited By 5. Elsevier B.V., 2018: pp. 9–20 (cit. on pp. 20, 66, 67).

[Neu18b] Neumaier, S., L. Thurnay, T.J. Lampoltshammer, et al.: 'Search, Filter, Fork, and Link Open Data: The ADEQUATe platform: Data- and community-driven quality improvements'. English. cited By 7. Association for Computing Machinery, Inc, 2018: pp. 1523–1526 (cit. on pp. 20, 77).

[Neu16] Neumaier, S., J. Umbrich, and A. Polleres: 'Automated quality assessment of metadata across open data portals'. English. *Journal of Data and Information Quality* (2016), vol. 8(1). cited By 108 (cit. on pp. 20, 49, 50, 77).

[Nik21] Nikiforova, A. and K. McBride: 'Open government data portal usability: A user-centred usability analysis of 41 open government data portals'. English. *Telematics and Informatics* (2021), vol. 58. cited By 47 (cit. on pp. 3, 21, 73).

[Nog21] Nogueras-Iso, J., J. Lacasta, M.A. Urena-Camara, et al.: 'Quality of Metadata in Open Data Portals'. English. *IEEE Access* (2021), vol. 9. cited By 6: pp. 60364–60382 (cit. on pp. 20, 77, 78).

[Noy19] Noy, N., M. Burgess, and D. Brickley: 'Google dataset search: Building a search engine for datasets in an open web ecosystem'. English. cited By 135. Association for Computing Machinery, Inc, 2019: pp. 1365–1375 (cit. on pp. 49, 50).

[Ojo20] Ojo, A. and O. Sennaike: 'Constructing Knowledge Graphs from Data Catalogues'. English. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2020), vol. 11969 LNCS. Ed. by Hung, D.V. and M. D'Souza. cited By 1: pp. 94–107 (cit. on pp. 20, 66, 67).

[Oli16a] Oliveira, M.I.S., L.E.R. De Alencar Oliveira, A.G. De Fátima Barros Lima, et al.: 'Enabling a unified view of open data catalogs'. English. Ed. by Hammoudi, S., L. Maciaszek, L. Maciaszek, M.M. Missikoff, O. Camp, J. Cordeiro, and J. Cordeiro. Vol. 2. cited By 7. SciTePress, 2016: pp. 230–239 (cit. on pp. 60, 62).

[Oli16b] Oliveira, M.I.S., H.R. De Oliveira, L.A. Oliveira, et al.: 'Open government data portals analysis: The Brazilian case'. English. Ed. by Kim, Y. and S.M. Liu. Vol. 08-10-June-2016. cited By 34. Association for Computing Machinery, 2016: pp. 415–424 (cit. on pp. 20, 54).

[Par09] Park, J.-R.: 'Metadata quality in digital repositories: A survey of the current state of the art'. English. *Cataloging and Classification Quarterly* (2009), vol. 47(3-4). cited By 134: pp. 213–228 (cit. on p. 77).

[Pes15] Pesce, V., A. Maru, P. Archer, et al.: 'Setting up a global linked data catalog of datasets for agriculture'. English. *Communications in Computer and Information Science* (2015), vol. 544. Ed. by Garoufallou, E., R.J. Hartley, and P. Gaitanou. cited By 4: pp. 357–368 (cit. on pp. 20, 60, 62).

[Pie18] Pietriga, E., H. Gözükan, C. Appert, et al.: 'Browsing linked data catalogs with LODAtlas'. English. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2018), vol. 11137 LNCS. Ed. by Bontcheva, K., D. Vrandecic, M.C. Suarez-Figueroa, M. Sabou, L. Kaffee, E. Simperl, V. Presutti, and Celino I. cited By 20: pp. 137–153 (cit. on p. 73).

[Por20] Portisch, J., O. Fallatah, S. Neumaier, et al.: 'Challenges of Linking Organizational Information in Open Government Data to Knowledge Graphs'. English. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2020), vol. 12387 LNAI. Ed. by Keet C.M., Dumontier M. cited By 0: pp. 271–286 (cit. on pp. 3, 21, 66, 67).

[Rei22] Reis, J.R., F. Bernadini, and J. Viterbo: 'A New Approach for Assessing Metadata Completeness in Open Data Portals'. English. *International Journal of Electronic Government Research* (2022), vol. 18(1). cited By 0 (cit. on pp. 10, 77).

[Roz12] Rozell, E., P. Fox, J. Zheng, et al.: 'S2S architecture and faceted browsing applications'. English. cited By 4. 2012: pp. 413–416 (cit. on pp. 73, 74).

[Rya22] Ryan, P., R. Brennan, and H.J. Pandit: 'DPCat: Specification for an Interoperable and Machine-Readable Data Processing Catalogue Based on GDPR'. English. *Information (Switzerland)* (2022), vol. 13(5). cited By 0 (cit. on pp. 1, 3, 10, 20, 60, 61, 63, 69, 70, 105).

[Sam23] Samarasinghe, S. and S. Lokuge: *Data democratization: Empowering employees for data-driven innovation*. English. cited By 0. IGI Global, 2023: pp. 155–183 (cit. on p. 21).

[Sch17] Scholz, R., N. Tcholtchev, P. Lämmel, et al.: 'A CKAN Plugin for data harvesting to the Hadoop distributed file system'. English. Ed. by Ferguson, D., V.M. Munoz, J. Cardoso, J. Cardoso, M. Helfert, and C. Pahl. cited By 3. SciTePress, 2017: pp. 19–28 (cit. on pp. 20, 49, 50, 60, 62).

[Sch18b] Scholz, R., N. Tcholtchev, P. Lämmel, et al.: 'From metadata catalogs to distributed data processing for smart city platforms and services: A study on the interplay of CKAN and hadoop'. English. *Communications in Computer and Information Science* (2018), vol. 864. Ed. by Helfert, M., J. Cardoso, C. Pahl, and D. Munoz V.M. anbd Ferguson. cited By 1: pp. 115–136 (cit. on pp. 60, 62).

[Scr22] Scrocca, M., A. Azzini, P. Bureš, et al.: 'Towards napDCAT-AP: Roadmap and Requirements for a Transportation Metadata Specification'. English. Ed. by Simsek, U., D. Chaves-Fraga, D. Chaves-Fraga, T. Pellegrini, and S. Vahdat. Vol. 3235. cited By 0. CEUR-WS, 2022 (cit. on pp. 20, 60, 61, 63).

[Sen04a] Sen, A.: 'Metadata management: Past, present and future'. English. *Decision Support Systems* (2004), vol. 37(1). cited By 69: pp. 151–173 (cit. on pp. 20, 66, 67, 69, 70).

[Sen18] Sennaike, O.A., M. Waqar, E. Osagie, et al.: 'Towards intelligent open data platforms: Discovering relatedness in datasets'. English. Vol. 2018-January. cited By 6. Institute of Electrical and Electronics Engineers Inc., 2018: pp. 414–421 (cit. on pp. 20, 49, 66, 67).

[Sha16b] Shanmugam, S. and G. Seshadri: 'Aspects of Data Cataloguing for Enterprise Data Platforms'. English. Ed. by Qiu, M. cited By 1. Institute of Electrical and Electronics Engineers Inc., 2016: pp. 134–139 (cit. on pp. 1, 10, 21, 66).

[Sha05] Shannon, C., D. Moore, K. Keys, et al.: 'The internet measurement data catalog'. English. Vol. 35. 5. cited By 25. 2005: pp. 97–100 (cit. on pp. 10, 20, 54–56, 66).

[Ško20] Škoda, P., D. Bernhauer, M. Nečaský, et al.: 'Evaluation Framework for Search Methods Focused on Dataset Findability in Open Data Catalogs'. English. Ed. by Indrawan-Santiago, M., E. Pardede, I.L. Salvadori, M. Steinbauer, I. Khalil, and G. Kotsis. cited By 2. Association for Computing Machinery, 2020: pp. 200–209.

[Ško19] Škoda, P., J. Klímek, M. Nečaský, et al.: 'Explainable Similarity of Datasets Using Knowledge Graph'. English. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2019), vol. 11807 LNCS. Ed. by Amato, G., C. Gennaro, V. Oria, and M. Radovanovic. cited By 2: pp. 103–110 (cit. on pp. 20, 66, 67).

[Sko19] Skopal, T., J. Klímek, and M. Neeàský: 'Improving findability of open data beyond data catalogs'. English. Ed. by Indrawan-Santiago, M., E. Pardede, I.L. Salvadori, M. Steinbauer, I. Khalil, and G. Anderst-Kotsis. cited By 2. Association for Computing Machinery, 2019 (cit. on pp. 20, 21, 66, 67).

[Šli21] Šlibar, B., D. Oreški, and N. Begičević Ređep: 'Importance of the Open Data Assessment: An Insight Into the (Meta) Data Quality Dimensions'. English. *SAGE Open* (2021), vol. 11(2). cited By 5 (cit. on p. 77).

[Spe22] Spezzatti, A., E. Kheradmand, K.K.G. Gupta, et al.: 'Note: Leveraging Artificial Intelligence to build a Data Catalog and support research on the Sustainable Development Goals'. English. Vol. Par F180472. cited By 0. Association for Computing Machinery, 2022: pp. 579–584.

[Tim23] Timmer, R.C., M. Mark, F.S. Khoo, et al.: 'NASA Science Mission Directorate Knowledge Graph Discovery'. English. cited By 1. Association for Computing Machinery, Inc, 2023: pp. 795–799 (cit. on pp. 20, 73).

[Tor19] Toro, J.F., D. Carrion, A. Albertella, et al.: 'CROSS-BORDER OPEN DATA SHARING: GIOCONDA PROJECT'. English. Ed. by Brovelli, M.A. and Florentina A.F. Andreea. Vol. 42. 4/W14. cited By 0. International Society for Photogrammetry and Remote Sensing, 2019: pp. 233–238 (cit. on pp. 60, 62).

[Tyg16] Tygel, A., S. Auer, J. Debattista, et al.: 'Towards Cleaning-Up Open Data Portals: A Metadata Reconciliation Approach'. English. cited By 11. Institute of Electrical and Electronics Engineers Inc., 2016: pp. 71–78 (cit. on pp. 20, 49, 50, 60, 62).

[Tzi21] Tzitzikas, Y., M. Pitikakis, G. Giakoumis, et al.: 'How Can a University Take Its First Steps in Open Data?' English. *Communications in Computer and Information Science* (2021), vol. 1355 CCIS. Ed. by Garoufallou E., Ovalle-Perandones M. cited By 5: pp. 155–167 (cit. on pp. 15, 54, 55, 60, 62).

[Urb22] Urbanek, S. and S. Schimmler: 'A Translation Service for Open Data Portals'. English. *eJournal of eDemocracy and Open Government* (2022), vol. 14(2). cited By 0: pp. 57–82 (cit. on pp. 1, 20, 49, 50).

[Wan17a] Wang, J., N. Car, B. Evans, et al.: 'Persistent identifier practice for big data management at NCI'. English. *Data Science Journal* (2017), vol. 16. cited By 1 (cit. on pp. 60, 62).

[Wan16] Wang, X., T. Tiropanis, and R. Tinati: 'WDFed: Exploiting cloud databases using metadata and RESTful APIs'. English. *Communications in Computer and Information Science* (2016), vol. 672. Ed. by Garoufallou, E., I.S. Coll, A. Stellato, and J. Greenberg. cited By 2: pp. 345–356 (cit. on pp. 60, 61).

[Won23] Won, H., J. Han, M.-S. Gil, et al.: 'SODAS: Smart Open Data as a Service for Improving Interconnectivity and Data Usability'. English. *Electronics (Switzerland)* (2023), vol. 12(5). cited By 0 (cit. on pp. 3, 20, 21).

[Yu16] Yu, M., Z. Jun, and L. Yan: 'Metadata scheme for open data in Chinese local government'. English. cited By 0. Computers and Industrial Engineering, 2016 (cit. on pp. 60, 62, 63).

[Zui16] Zuiderwijk, A., M. Janssen, and I. Susha: 'Improving the speed and ease of open data use through metadata, interaction mechanisms, and quality indicators'. English. *Journal of Organizational Computing and Electronic Commerce* (2016), vol. 26(1-2). cited By 50: pp. 116–146 (cit. on pp. 22, 73, 77).

A DIVA Screenshots

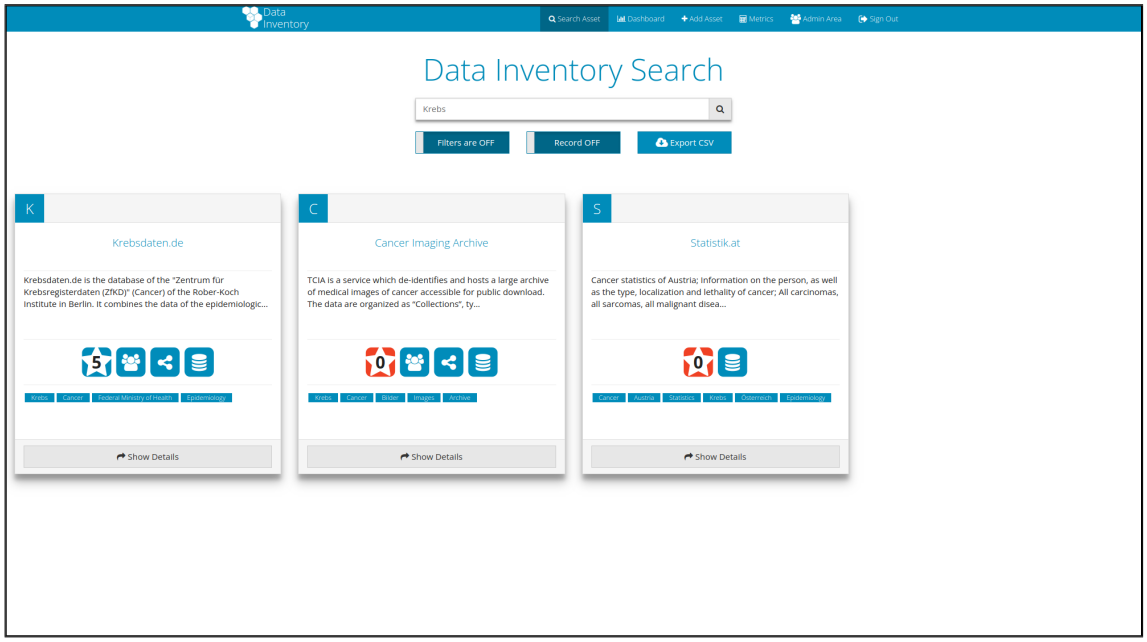


Figure A.1: DIVA 1.0: Search functionality of the data catalog

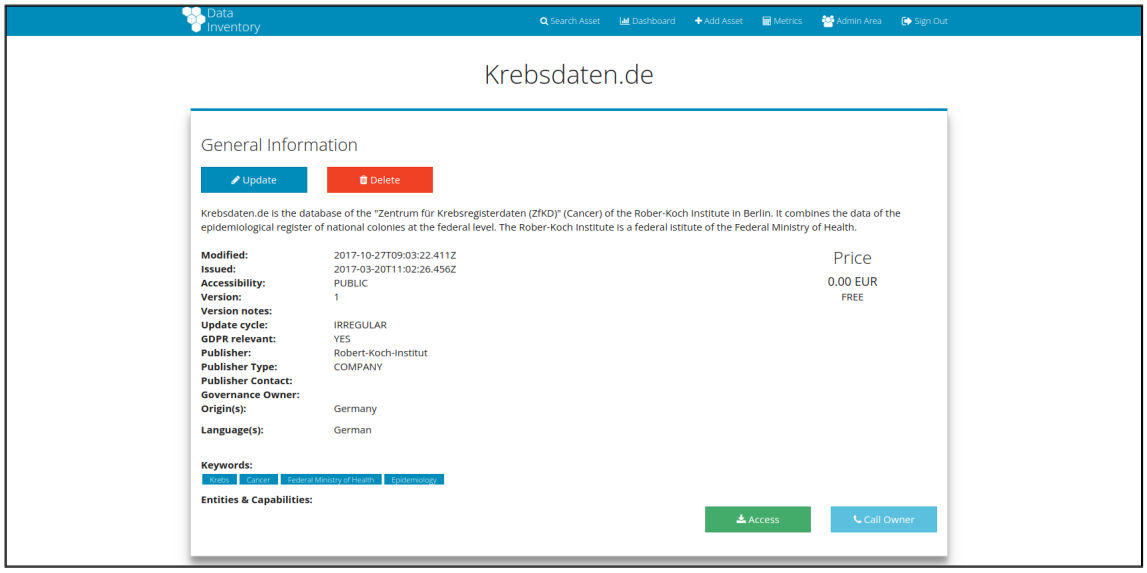


Figure A.2: DIVA 1.0: Details view

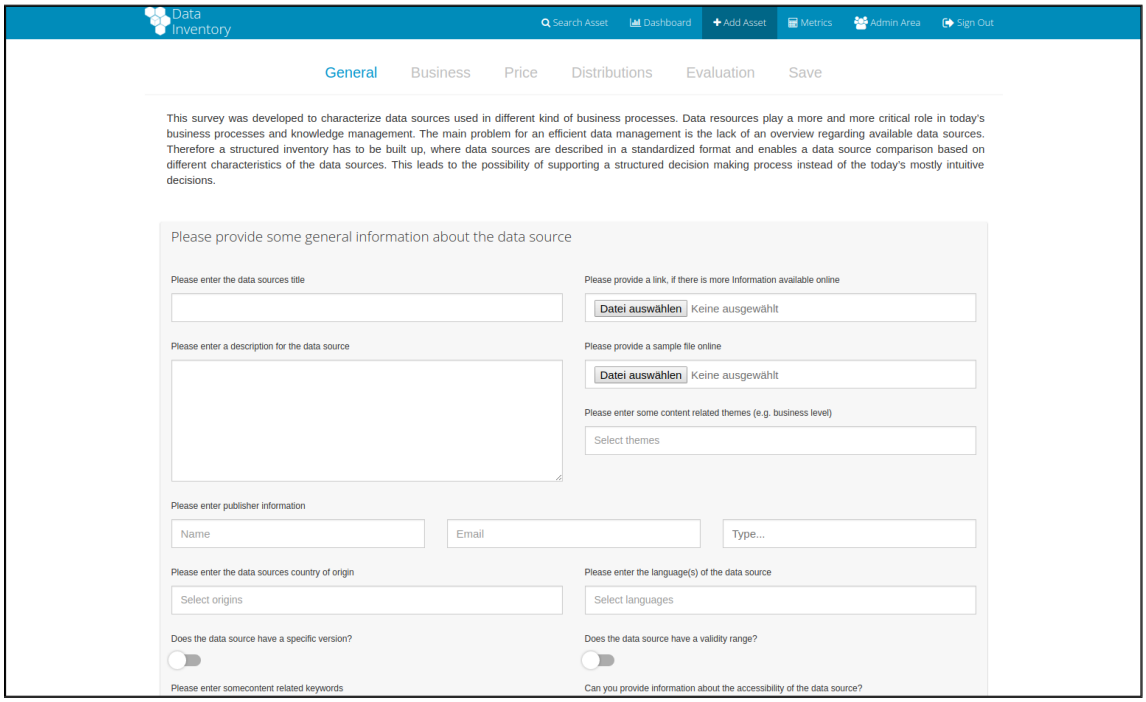


Figure A.3: DIVA 1.0: Data inventory form

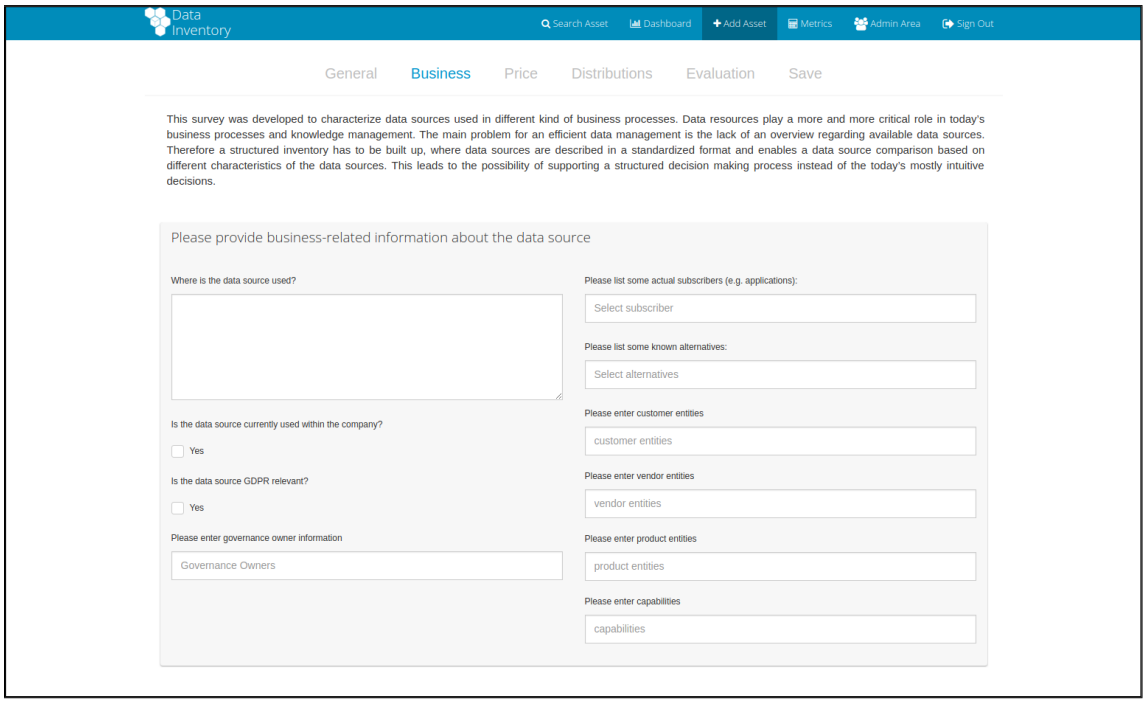


Figure A.4: DIVA 1.0: Data inventory business form

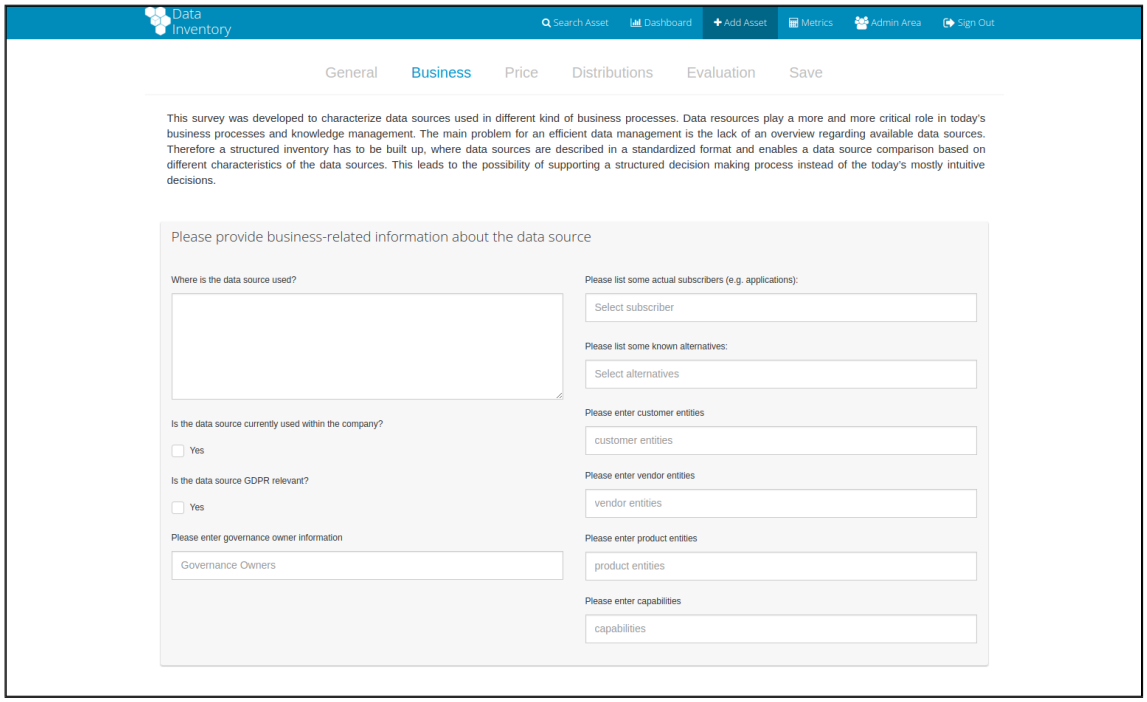


Figure A.5: DIVA 1.0: Data inventory price form

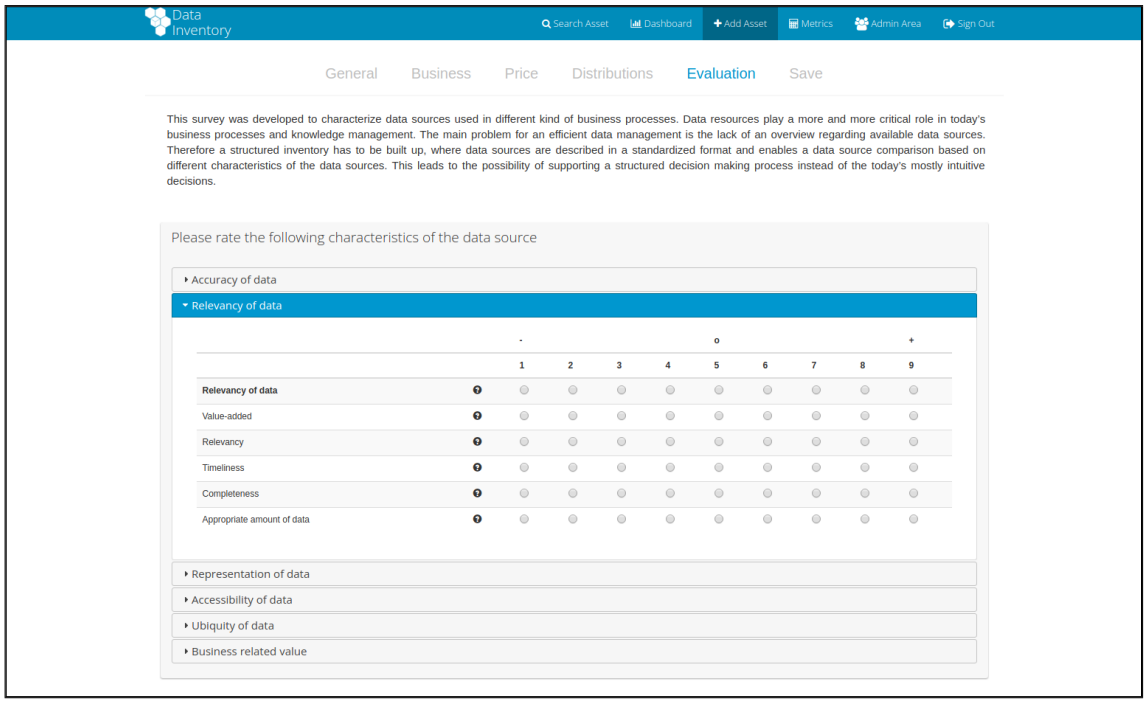


Figure A.6: DIVA 1.0: Data evaluation questionnaire

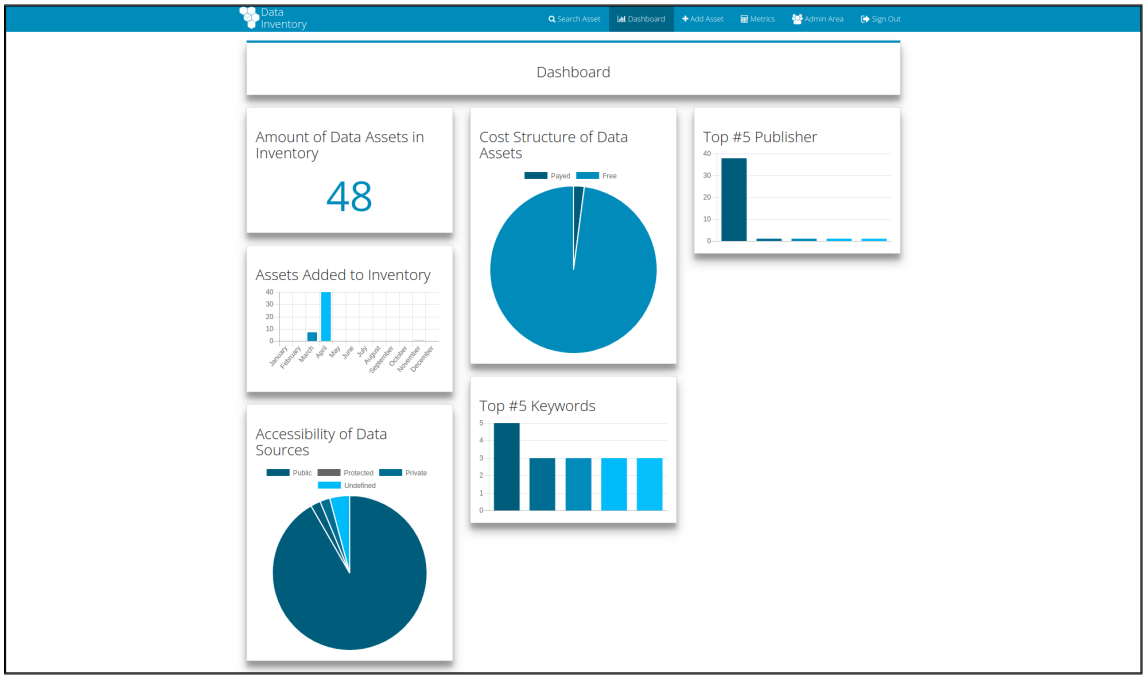


Figure A.7: DIVA 1.0: Management dashboard

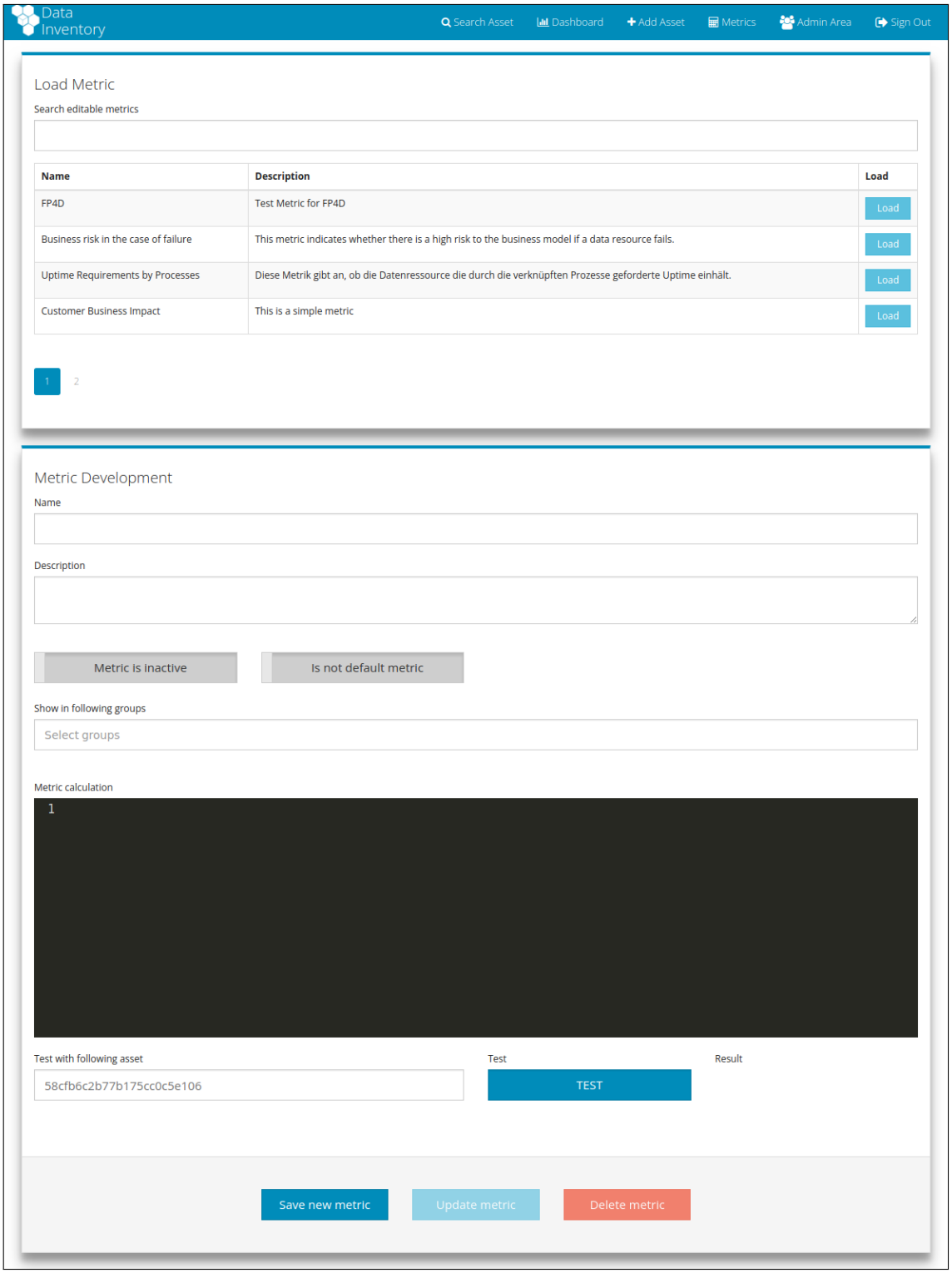
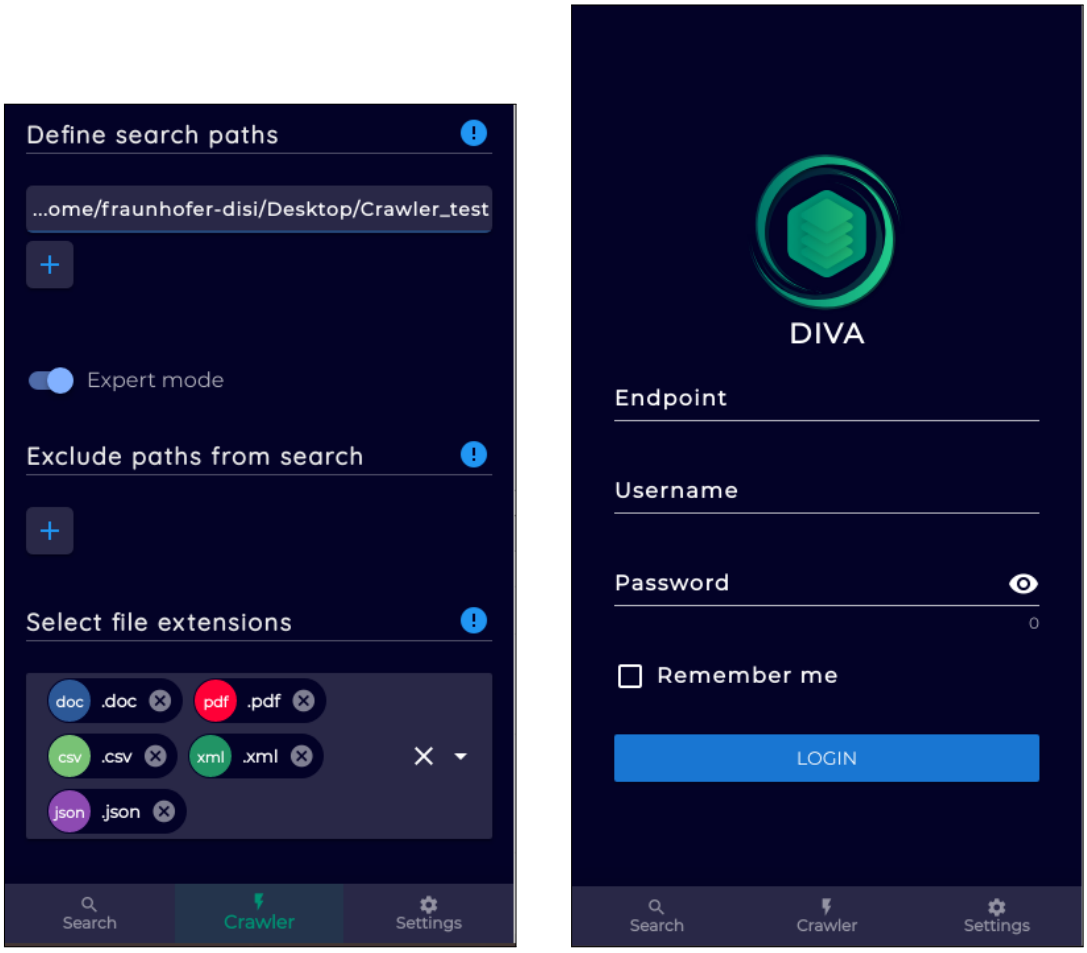


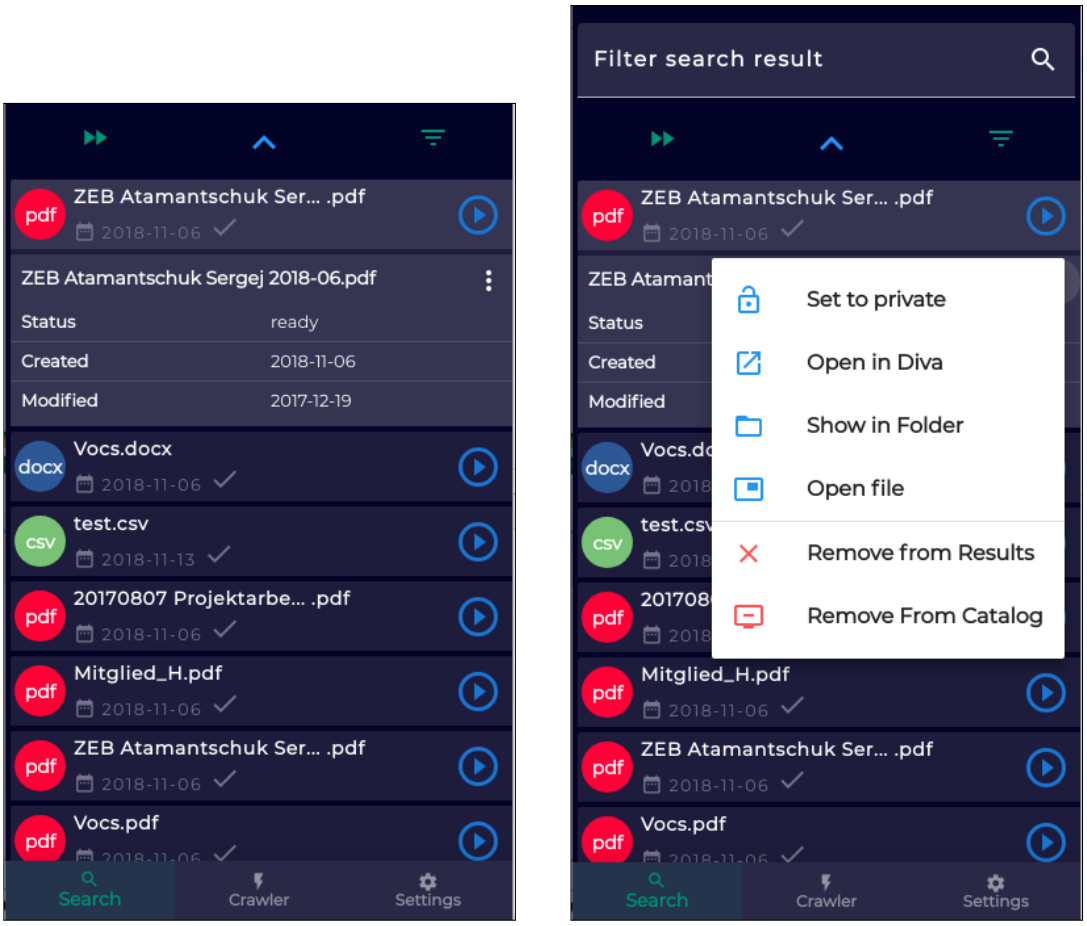
Figure A.8: DIVA 1.1: Custom metrics

Figure A.9: DIVA 1.1: Data source evaluation results



(a) Data Asset Crawler configuration **(b)** Data Asset Crawler login

Figure A.10: Data Asset Crawler screenshots (1)



(a) Data Asset Crawler crawling results **(b)** Data Asset Crawler results menu

Figure A.11: Data Asset Crawler screenshots (2)

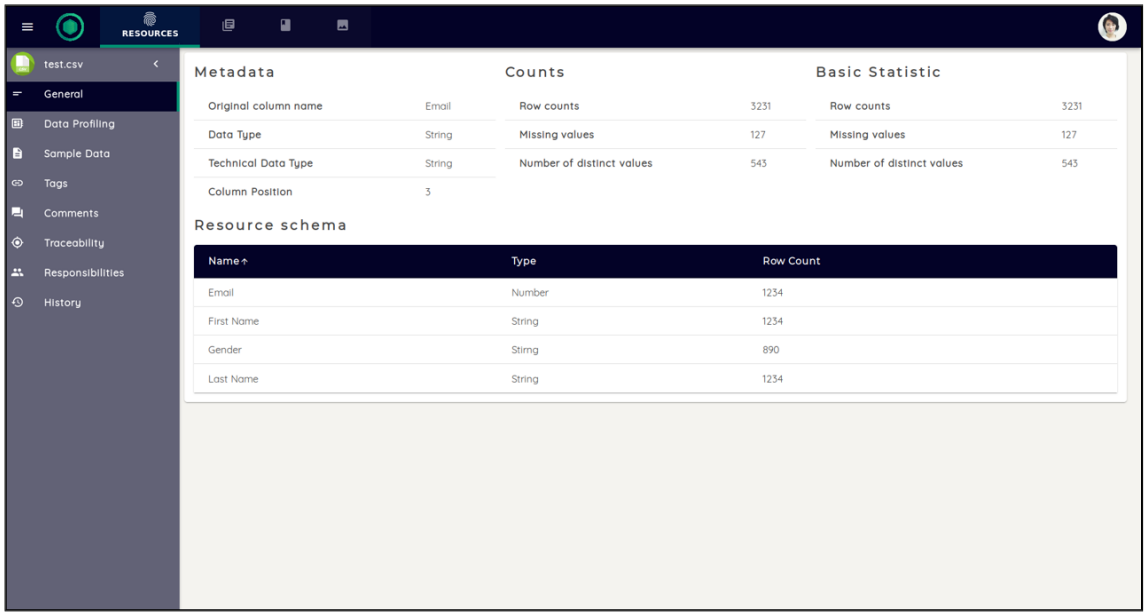


Figure A.12: DIVA 2.1: Metadata representation of tabular data

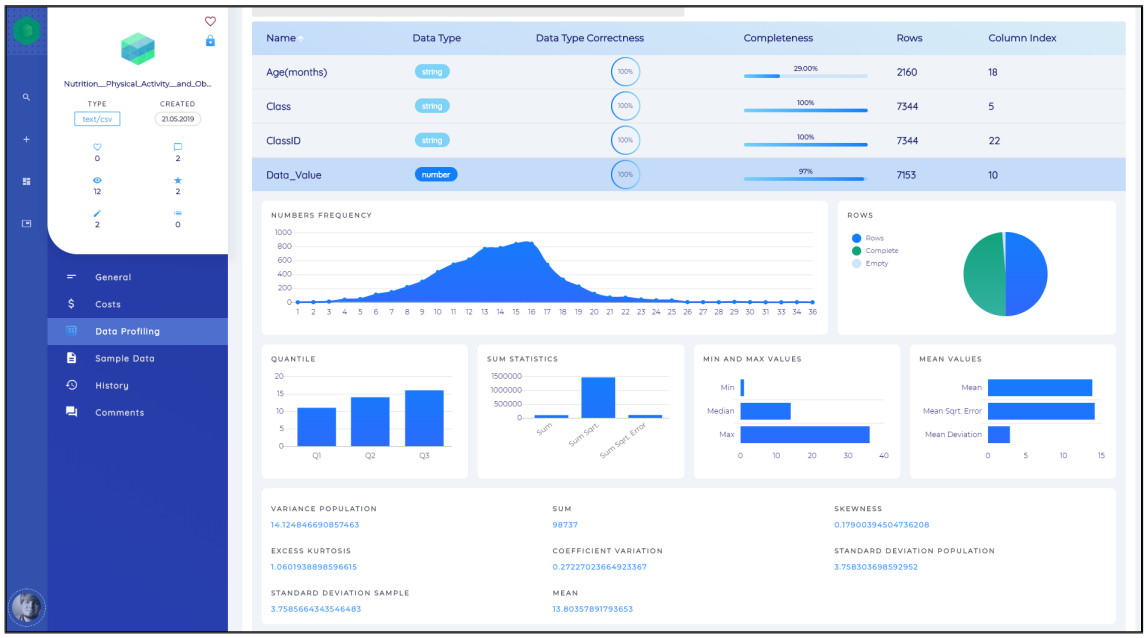


Figure A.13: DIVA 3.0: Metadata representation of tabular data

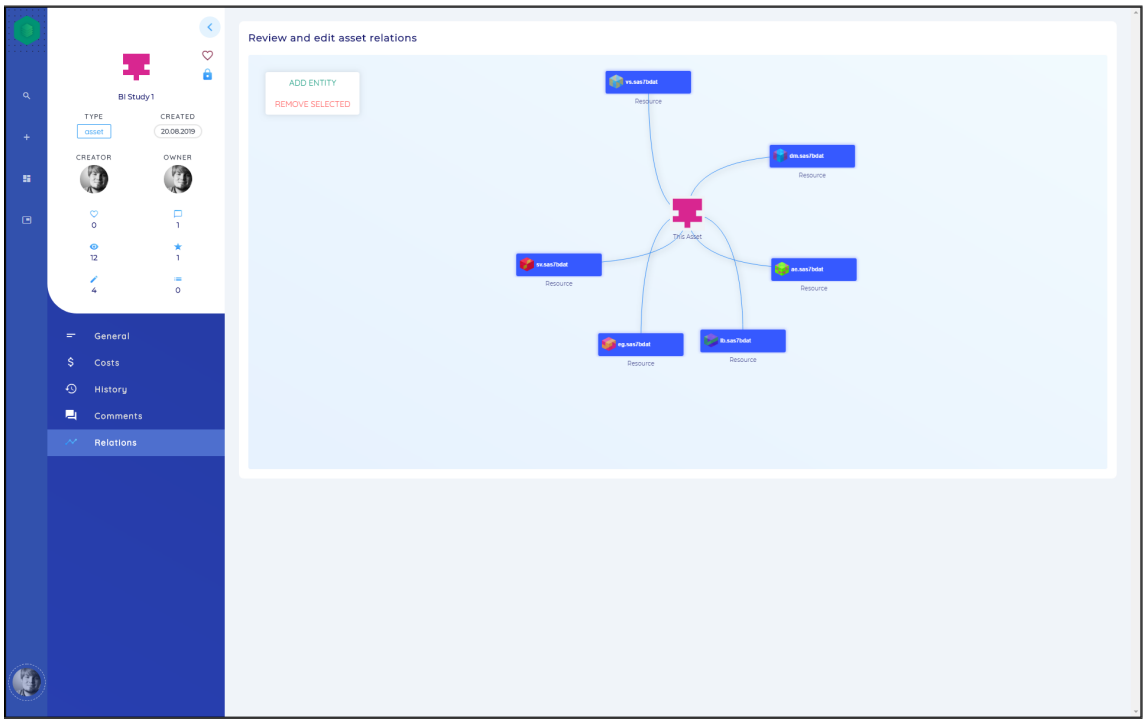


Figure A.14: DIVA 3.0: Visualization of the data network

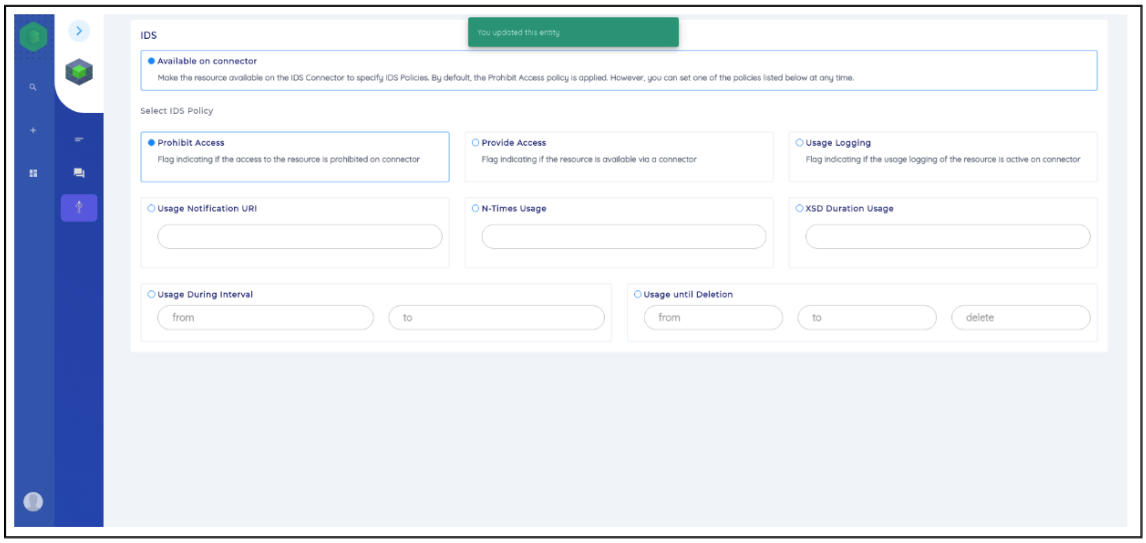


Figure A.15: DIVA 3.0: Data Space Connector usage policy configuration

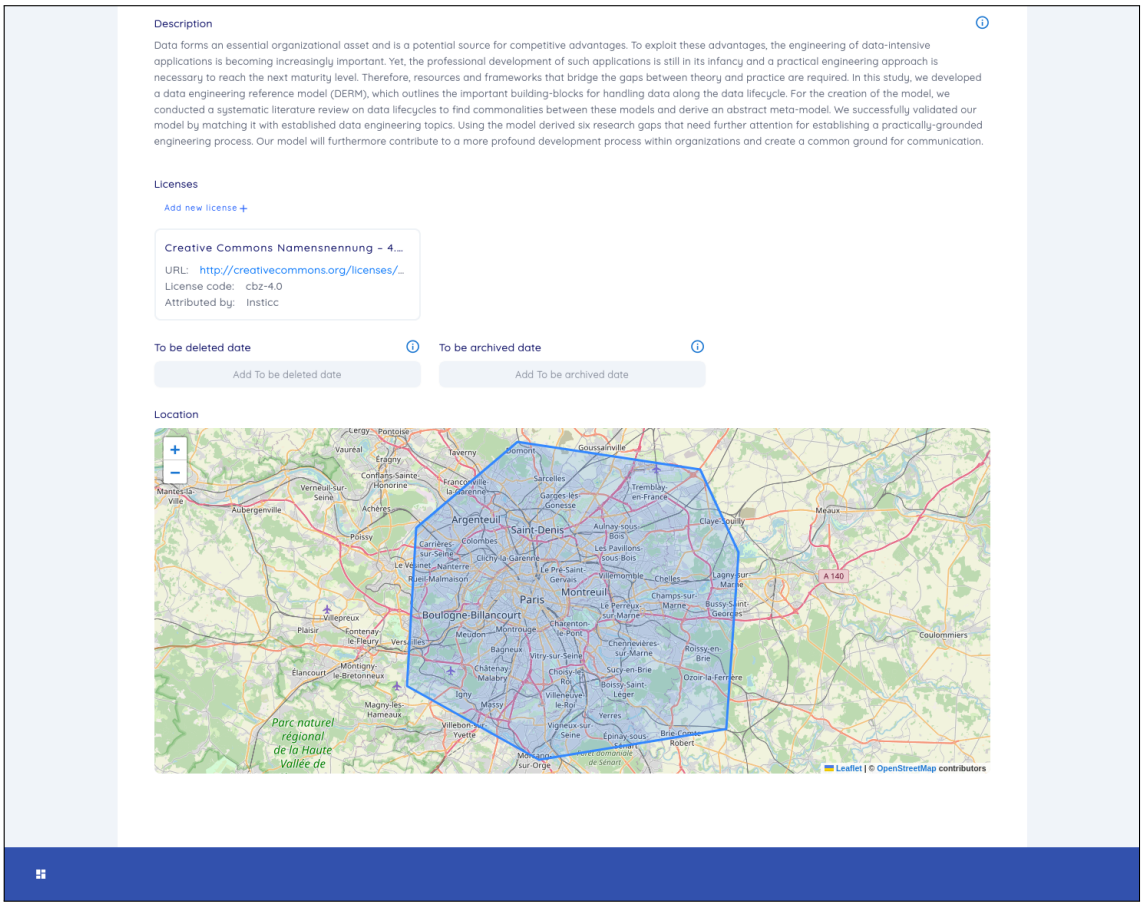


Figure A.16: DIVA 4.0: Details view

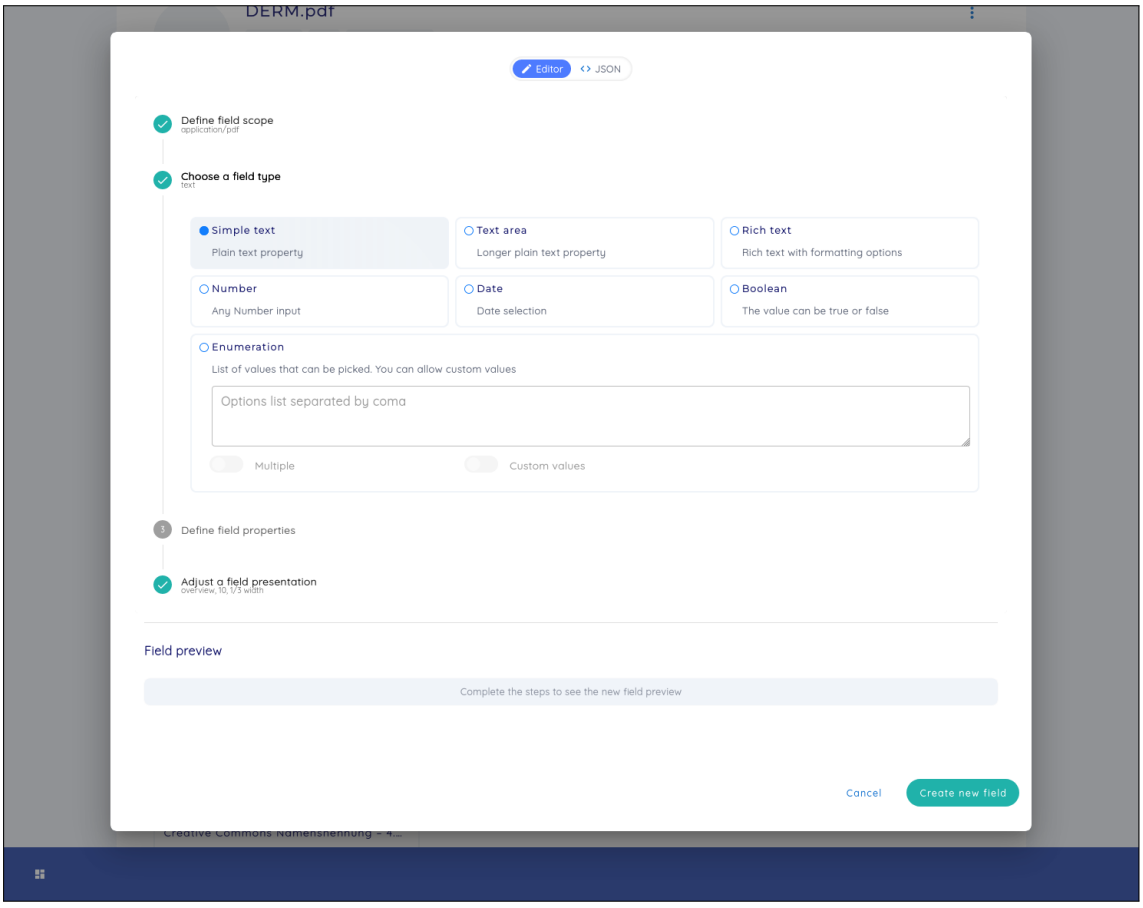


Figure A.17: DIVA 4.0: A form for creating new metadata fields

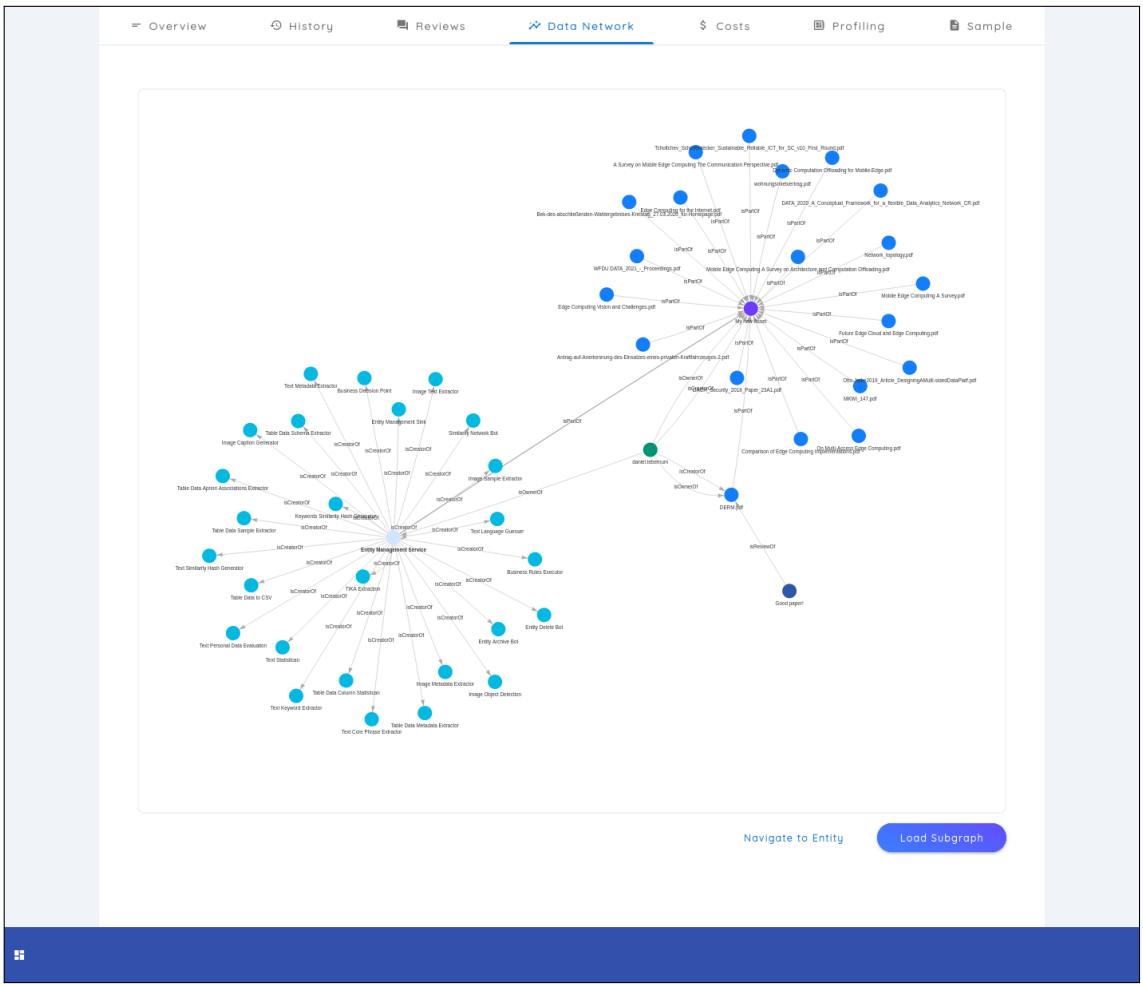


Figure A.18: DIVA 4.0: Data network visualization

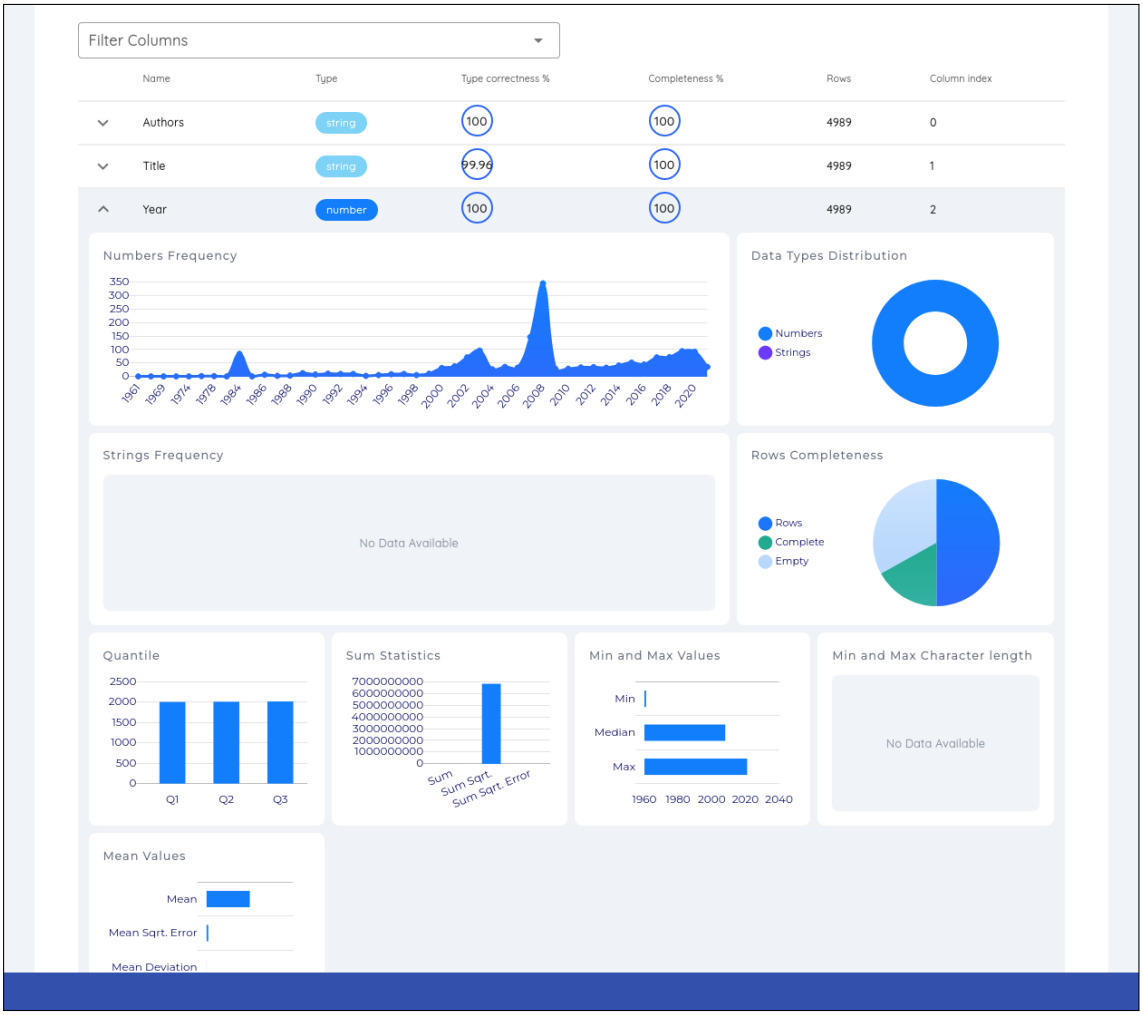


Figure A.19: DIVA 4.0: Metadata representation of tabular data

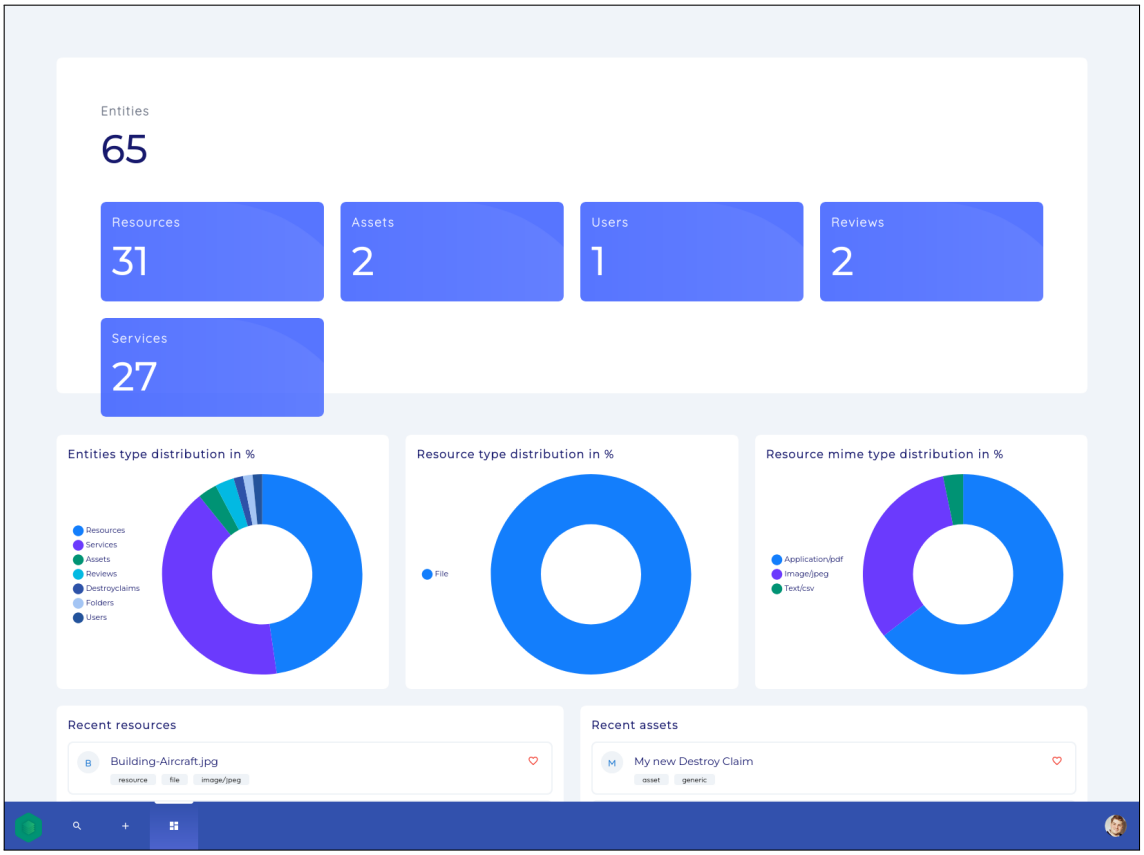


Figure A.20: DIVA 4.0: Management dashboard

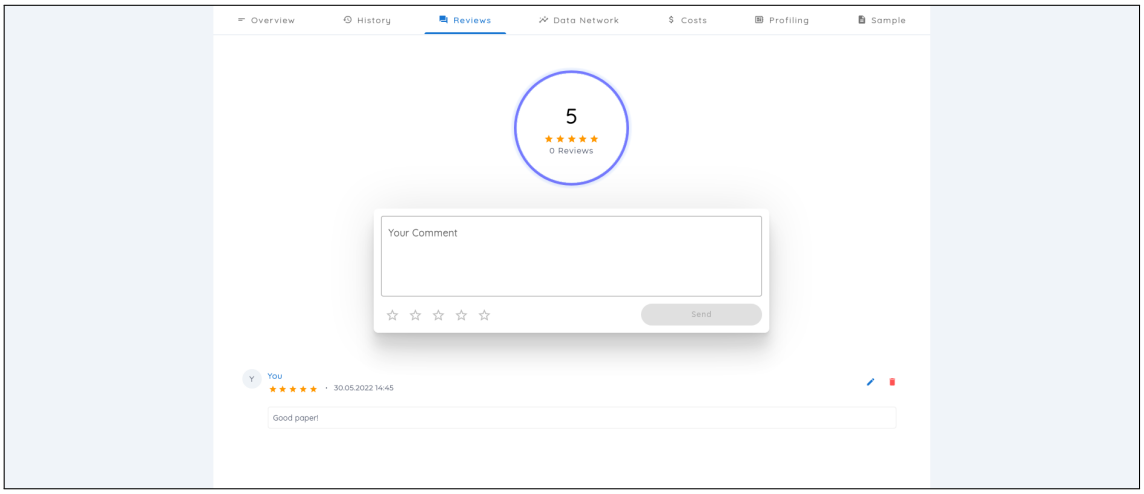


Figure A.21: DIVA 4.0: Review system

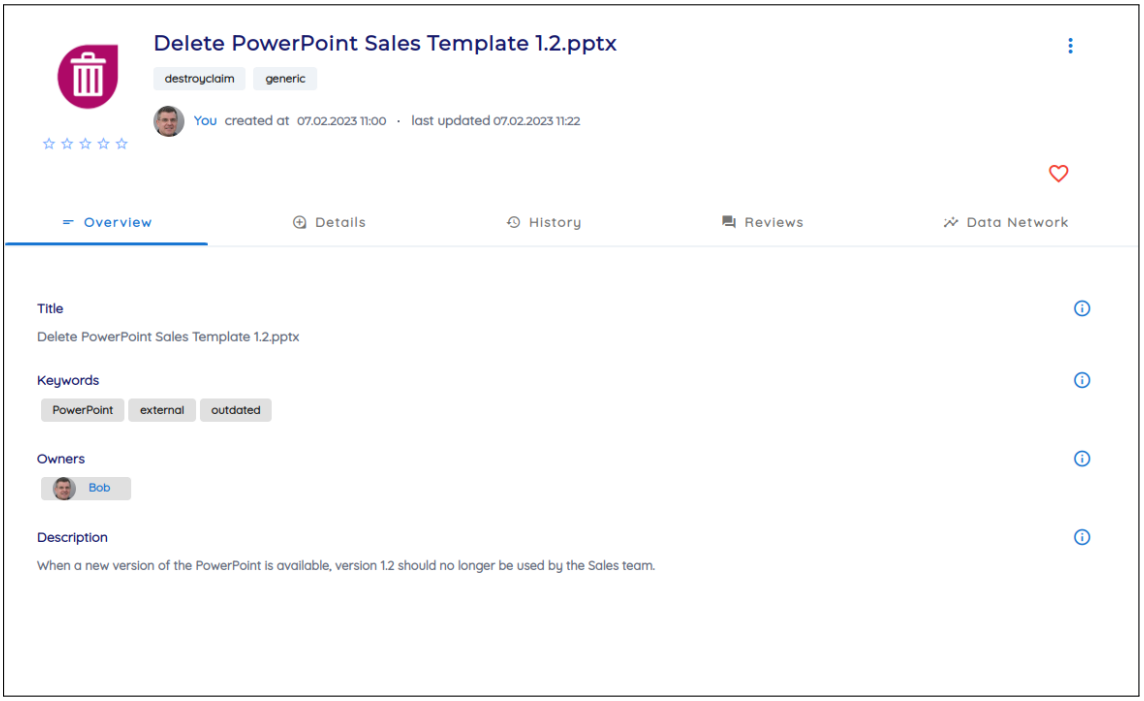


Figure A.22: DIVA 4.1: Destroy Claim overview

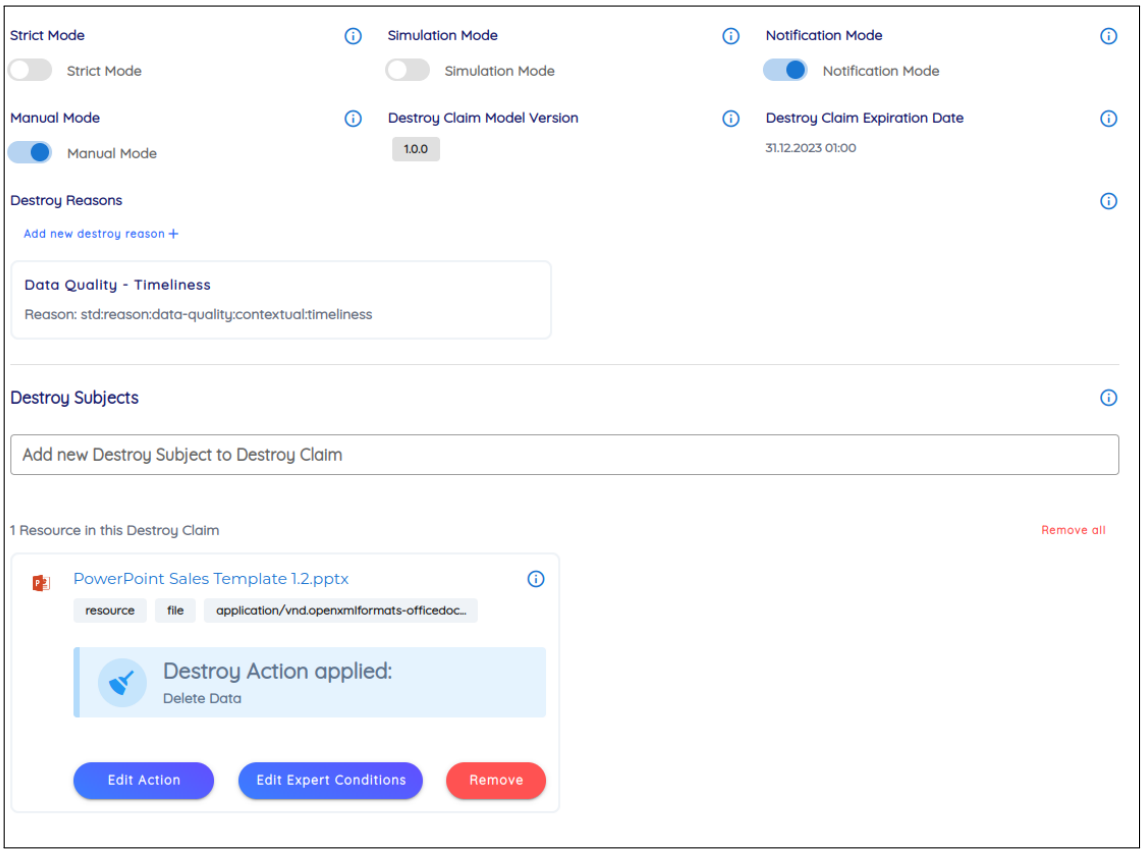


Figure A.23: DIVA 4.1: Destroy Claim details part 1

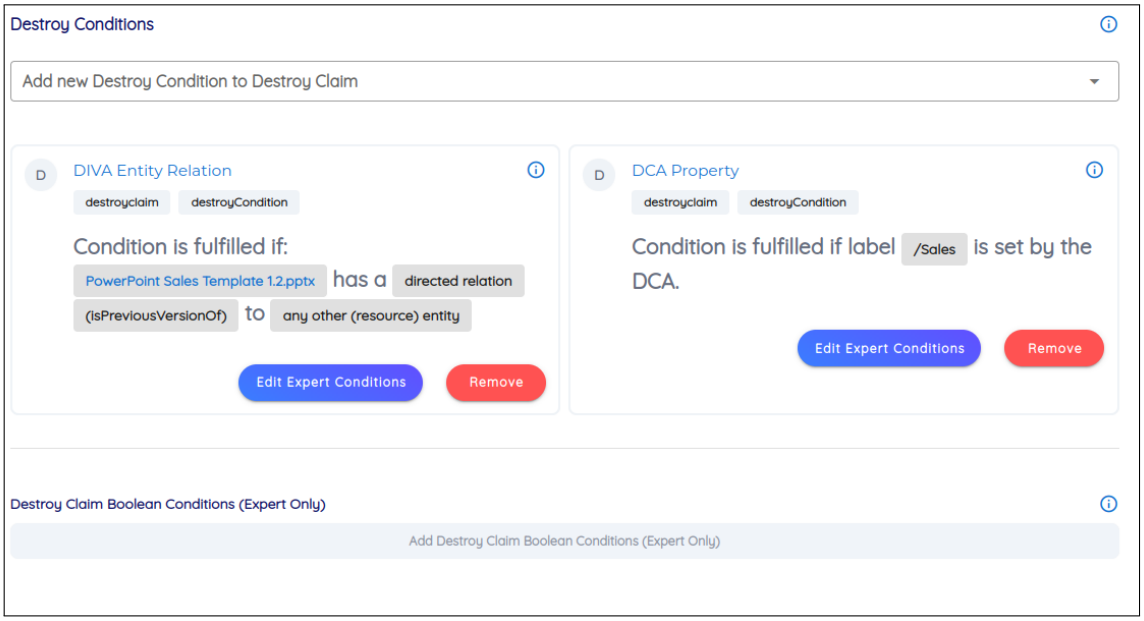


Figure A.24: DIVA 4.1: Destroy Claim details part 2

B Evaluation Infomaterial

Design Principles for Data Catalogs - Infomaterial

Daniel Tebernum

In this research project, design principles and design features for data catalogs are investigated. Design principles and design features are an important form of design knowledge and in this case are intended to help practitioners design, implement, or select better data catalogs. Over a period of six years, the open source data catalog DIVA (`https://github.com/FraunhoferISST/diva/`) was developed iteratively within the scope of various industry and research projects. The insights gained there were condensed into seven abstract design principles. Each of the design principles is concretized by a set of design features. Please familiarize yourself with the design principles and design features by reading them carefully. Afterwards we will give you access to a short questionnaire. We will then also ask you in person about your opinions and ideas. This will take about an hour of your time.

Design principle title	**DP1: Principle of Automation**
Aim, implementer, and user	To efficiently and cost-effectively provide an up-to-date and comprehensive data catalog inventory with accurate, error-free, and high-quality content (aim) for its users (user).
Mechanism	Automate as many processes as possible that are necessary for the seamless operation of the data catalog.
Rationale	Due to the extensive amount of existing data, it is impossible to manually maintain the data catalog content efficiently.
Design Features	To implement DP1, ... • provide mechanisms to automatically inventory relevant data and other entities. • provide mechanisms that automatically discover relevant metadata for inventoried data.

Design principle title	**DP2: Principle of Flexibility**
Aim, implementer, and user	To provide a data catalog that can be adapted to different contexts, environments, and future challenges (aim) by the users (user).
Mechanism	Allow customization of components and features.
Rationale	Due to the highly individual requirements demanded of a data catalog, there is no one-fits-all solution. Having the data catalog code itself customized creates unnecessary costs, creates artificial hurdles, and delays the adoption of a data catalog.
Design Features	To implement DP2, ... • provide mechanisms for users to store new metadata attributes in the data catalog. • provide mechanisms for users to generate their own metrics based on the metadata stored in the data catalog. • provide mechanisms for users of the data catalog to fine-tune access restrictions to their needs. • provide mechanisms for users to customize workflows to their needs.

Design principle title	**DP3: Principle of Interoperability**
Aim, implementer, and user	For the data catalog to seamlessly cooperate with other data catalogs and other systems in general (aim).
Mechanism	Standardize, document, and publish the metadata models, APIs, processes, and source code.
Rationale	Data catalogs must be able to embed themselves in existing technology landscapes and understand many different systems to be used reasonably.
Design Features	To implement DP3, ... • standardized metadata models should be used. • standardized API documentation should be provided. • make the data catalog available as open-source software and use established open-source software

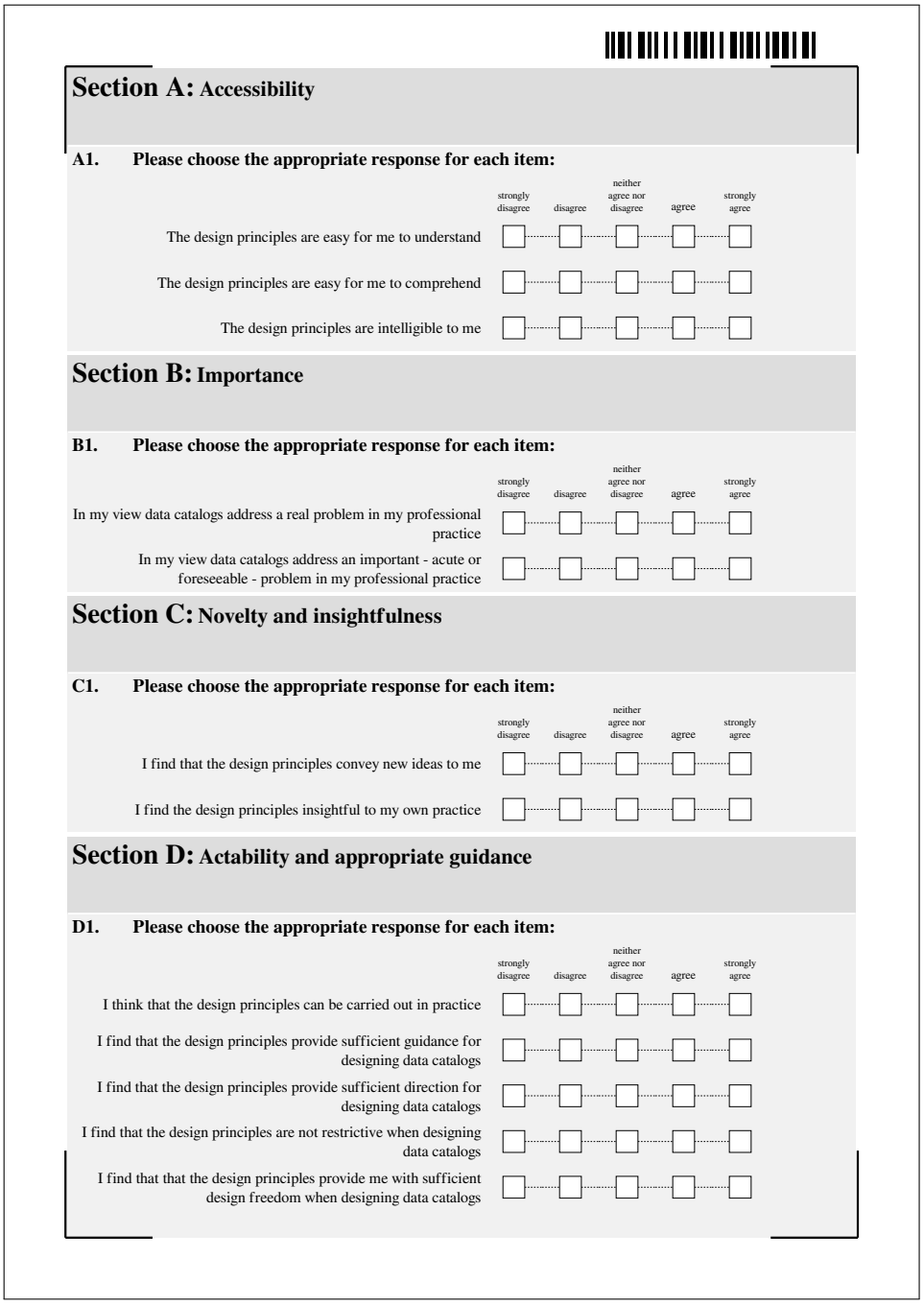
Design principle title	**DP4: Principle of Context**
Aim, implementer, and user	For the users of the data catalog (user) to gain a better understanding and deeper insight into the inventoried data (aim).
Mechanism	Provide data-system relationships, data lineage, data linkage, business context, and technical context by enriching metadata, connecting data to related entities in the data catalog, and answering questions regarding the what, who, where, when, why, and how.
Rationale	Context allows for faster, better, and more accessible data understanding.
Design Features	To implement DP4, ... • extend the metadata model by as many context-related attributes as possible. • establish relationships between data and other entities stored in the data catalog, such as users, publishers, or projects, and thereby create a data network.

Design principle title	**DP5: Principle of Data Life Cycle Management**
Aim, implementer, and user	For the users of the data catalog (user) to share and gain necessary operational knowledge about the data (aim).
Mechanism	Equip the data catalog with data life cycle information and associated management functions.
Rationale	The availability of data life cycle information enables users of the data catalog to assess operational consequences immediately.
Design Features	To implement DP5, ... • provide usage policies for the inventoried data in the data catalog to prevent unauthorized access, mishandling or to ensure legal and regulatory compliance. • enable end-of-life data management in one's data catalog to, among other objectives, ensure regulatory and legal compliance, prevent the proliferation of poor-quality data, protect security-critical data, save costs, or destroy broken data.

Design principle title	**DP6: Principle of Visualization**
Aim, implementer, and user	To allow the data catalog user (user) to identify relationships and patterns regarding data and metadata more easily and quickly (aim).
Mechanism	Provide visualizations of non-trivial aspects and tailor them to fit different use cases and roles.
Rationale	Humans can recognize connections and patterns better and faster when visualizing data. Data catalogs contain a lot of metadata that a user needs to process.
Design Features	To implement DP6, ... • provide visualizations for non-trivial metadata. • visualize the relations between data and other entities stored in the data catalog

Design principle title	**DP7: Principle of Data Assessment**
Aim, implementer, and user	To allow data catalog users (user) to make more informed decisions and to increase confidence (aim) in the data.
Mechanism	Provide data-related assessments, e.g., for quality, risks, integration capability, and others.
Rationale	An assessment of the data is necessary so that users of the data catalog can gain insight into the quality of the data, whether it poses a risk, complies with data governance rules, or even whether integration is easily possible. Gathering this information in one place gives users a better overview and saves time.
Design Features	To implement DP7, ... • provide quality metrics for the data inventoried and metadata stored in the data catalog. • provide a review system through which users can share their experiences with the inventoried data. • provide risk metrics so that users of the data catalog can make informed decisions based on them and take countermeasures when risks are high.

C Evaluation Questionnaire



Section A: Accessibility

A1. Please choose the appropriate response for each item:

	strongly disagree	disagree	neither agree nor disagree	agree	strongly agree
The design principles are easy for me to understand	☐	☐	☐	☐	☐
The design principles are easy for me to comprehend	☐	☐	☐	☐	☐
The design principles are intelligible to me	☐	☐	☐	☐	☐

Section B: Importance

B1. Please choose the appropriate response for each item:

	strongly disagree	disagree	neither agree nor disagree	agree	strongly agree
In my view data catalogs address a real problem in my professional practice	☐	☐	☐	☐	☐
In my view data catalogs address an important - acute or foreseeable - problem in my professional practice	☐	☐	☐	☐	☐

Section C: Novelty and insightfulness

C1. Please choose the appropriate response for each item:

	strongly disagree	disagree	neither agree nor disagree	agree	strongly agree
I find that the design principles convey new ideas to me	☐	☐	☐	☐	☐
I find the design principles insightful to my own practice	☐	☐	☐	☐	☐

Section D: Actability and appropriate guidance

D1. Please choose the appropriate response for each item:

	strongly disagree	disagree	neither agree nor disagree	agree	strongly agree
I think that the design principles can be carried out in practice	☐	☐	☐	☐	☐
I find that the design principles provide sufficient guidance for designing data catalogs	☐	☐	☐	☐	☐
I find that the design principles provide sufficient direction for designing data catalogs	☐	☐	☐	☐	☐
I find that the design principles are not restrictive when designing data catalogs	☐	☐	☐	☐	☐
I find that that the design principles provide me with sufficient design freedom when designing data catalogs	☐	☐	☐	☐	☐

Section E: Effectiveness

E1. Please choose the appropriate response for each item:

	strongly disagree	disagree	neither agree nor disagree	agree	strongly agree
I believe that the design principles can help design data catalogs in practice	☐	☐	☐	☐	☐
I find the design principles useful for designing data catalogs in practice	☐	☐	☐	☐	☐
I believe that the design principles can help design data catalogs that improve the performance of its users	☐	☐	☐	☐	☐
I believe that the design principles can help design data catalogs that improve the productivity of its users	☐	☐	☐	☐	☐
I believe that the design principles can help design data catalogs that improve the effectiveness of its users	☐	☐	☐	☐	☐
I believe that the design principles can help design data catalogs that improve the quantity of work of its users	☐	☐	☐	☐	☐
I believe that the design principles can help design data catalogs that improve the quality of work of its users	☐	☐	☐	☐	☐
I believe that the design principles can help design data catalogs that improve the innovativeness of an organization/company	☐	☐	☐	☐	☐
I believe that the design principles can help design data catalogs that improve the reputation of an organization/company	☐	☐	☐	☐	☐
I believe that the design principles can help design data catalogs that improve the job morale of an organization/company	☐	☐	☐	☐	☐

D Expert Discussion Notes

D.1 Expert E1

Table D.1: Discussion notes E1

Context	Software Engineer in Research (E1)	Codes
DP1	There is no problem in understanding DP1. Both design principles and design features are clearly described. Data marketplace operators see this as an essential issue. Automation is realistic but challenging because many technologies need to be connected. Perhaps the price of data can be determined automatically. That would help to reduce the amount of work. Prices could adjust depending on the market situation.	Accessibility(+) Importance(+) Novelty(∘) Actability(+) Effectiveness(+)
DP2	There is no problem in understanding DP2. Flexibility is vital for many types of software. Possible new design feature: The user interface can be customized for different roles. This facilitates the work with the data catalog. However, flexibility is also time-consuming to implement and is avoided if it offers no added value. Due to the complexity of data catalogs, making them more flexible makes sense. In the case of data marketplaces, flexibility would also make less sense for end users than providers. New data offerings must be added without the need to customize the marketplace software code. This happens regularly, for example, when adding new metadata types. One cannot always customize the code.	Accessibility(+) Importance(+) Novelty(-) Actability(+) Effectiveness(+)

Continued on next page

Table D.1 – continued from previous page

DP3	There is no problem in understanding DP3. Customers want specific standards for the data marketplace to keep it compatible with the rest of the ecosystem. E.g., IoT standard models must be used in metadata. This is the only way to collaborate effectively at all. Interoperability is an important but known problem. Nowadays, no matter what software one builds, one already thinks of this. It is realistic to implement this when one has a limited quantity of technologies.	Accessibility(+) Importance(+) Novelty(-) Actability(+) Effectiveness(∘)
DP4	There is no problem in understanding DP4. Not everything can be covered easily. Data lineage, for example, is only possible if all data-holding systems are monitored. Automation is needed. However, such information is very important for customers. It decides whether data is purchased or not. The principle is on such a high level that even the design features only give a rough idea of what should be included in terms of context. However, it is also highly dependent on the data, so putting the "what" into a principle is hard.	Accessibility(+) Importance(+) Novelty(∘) Actability(-) Effectiveness(∘)
DP5	There is no problem in understanding DP5. This is something new in the data catalog and especially in the data marketplace domain. Relevant for data marketplaces for managing the expiration of data.	Accessibility(+) Importance(+) Novelty(+) Actability(∘) Effectiveness(∘)
DP6	There is no problem in understanding DP6. One of the most important points is that using the data marketplace is much easier. It is clear what has to be done. In detail, however, one has to look at how to visualize. Nevertheless, this kind of detail does not need to be part of the design principle.	Accessibility(+) Importance(+) Novelty(∘) Actability(∘) Effectiveness(+)

Continued on next page

Table D.1 – continued from previous page

DP7	There is no problem in understanding DP7. Data marketplaces should have mechanisms for assessment (e.g., data quality). Buyers should be able to see how good the quality is. This is a concrete requirement of stakeholders. Data assessment is complex due to the heterogeneity of the data. Requirements regarding the assessment should be adapted step by step. This also saves an enormous amount of time for the user.	Accessibility(+) Importance(∘) Novelty(∘) Actability(∘) Effectiveness(+)
Overall	The legal context seems to be missing. Under which licenses are the data offered? Does the data contain personal information? Even though it falls under DP4 or DP5, it is important to mention. It is not a principle deficit but possibly a feature deficit.	Principle Deficit

D.2 Expert E2

Table D.2: Discussion notes E2

Context	Lead Software Engineer in Automotive (E2)	Codes
DP1	Important core topic. Much work is done in this area in general. The data catalog has to automate most of its tasks, as it would be impossible to do this manually. Automation is a continuous task. For DP1, it is clear what it means and what needs to be done to implement it.	Accessibility(+) Importance(+) Novelty(-) Actability(+) Effectiveness(∘)
DP2	Flexibility is important and justifiable due to the complexity of individual use cases. For DP2, it is clear what it means and what needs to be done to implement it.	Accessibility(+) Importance(+) Novelty(∘) Actability(+) Effectiveness(∘)
DP3	It sounds very common that one should ensure interoperability. On the other hand, it is especially important for something like a data catalog, especially if one wants to implement DP1. One has to be able to rely on quite a few things. DP1 and DP3 complement each other well. With both, one can ensure that the data catalog is well populated. For DP3, it is clear what it means and what needs to be done to implement it.	Accessibility(+) Importance(+) Novelty(-) Actability(+) Effectiveness(∘) Priciple Redundancy
DP4	This is very important as without context, the inventoried data is useless. For DP4, it is clear what it means and what needs to be done to implement it.	Accessibility(+) Importance(+) Novelty(∘) Actability(+) Effectiveness(∘)
DP5	DP5 is easy to understand. It is important to have operational knowledge from the beginning. If one plans a data catalog without operational knowledge in mind, the data catalog will be obsolete or useless.	Accessibility(+) Importance(+) Novelty(∘) Actability(∘) Effectiveness(+)

Continued on next page

Table D.2 – continued from previous page

DP6	DP6 is easy to understand. It is only important if the data catalog has a user interface. Some data catalogs do not have a UI.	Accessibility(+) Importance(∘) Novelty(∘) Actability(∘) Effectiveness(∘)
DP7	DP7 is easy to understand. Important but not for every use case. Specific data catalogs may not need to do assessments. It is actionable, but the number of different data types and domains makes implementation continuous.	Accessibility(+) Importance(∘) Novelty(∘) Actability(+) Effectiveness(∘)
Overall	The DPs can be focused one at a time. E.g. first, make context and interoperability. Next, do automation and life cycle management. There are none of the deficiencies mentioned. At this high level, it is hard to think about additional design principles or phenomena not somehow covered by what is already there. A decision tree for design principles and design features would be good ("What should I do?"). Become more concrete for practitioners. Design features are a good start but need to be more refined and comprehensive.	

D.3 Expert E3

Table D.3: Discussion notes E3

Context	Applied Researcher and Software Engineer at a Cloud Provider (E3)	Codes
DP1	This DP is understood. This is one of the most important aspects to get right. There are unambiguous requirements for the degree of automation. Due to the mass of data in cloud systems, a data catalog must handle almost everything automatically. If one only has to make the most important other systems known to the catalog and everything else runs by itself from then on, that would be good. It would save much time and make everything more manageable.	Accessibility(+) Importance(+) Novelty(∘) Actability(∘) Effectiveness(+)
DP2	This DP is understood. It is imperative, as there is no one size fits all. In the search for data catalogs, flexibility is put behind open source when in doubt because FOSS gives much flexibility when one has software developers. An important aspect is to connect new types of data sources in the future. That is lacking in the design features. The DP is realistic to implement but expensive.	Accessibility(+) Importance(+) Novelty(∘) Actability(+) Effectiveness(∘) Principle Deficit
DP3	The DP is understood. Interoperability can mean that the data catalog can be easily exchanged. This is more of an academic problem and is not so easy and needed in practice. However, open source software is a critical aspect. The company is looking for a data catalog to inventory data hosted in its cloud systems. It must be an open-source solution so that it can be adapted in case of doubt. Nevertheless, this is true for a lot of software and thus has been around for a while.	Accessibility(+) Importance(+) Novelty(-) Actability(∘) Effectiveness(∘)

Continued on next page

Table D.3 – continued from previous page

DP4	The design principle is formulated well. It can be important depending on what kind of data catalog one has. In most cases, context knowledge will help users and machines. Providing context is nothing new. Many domain-specific models already do that. It is feasible, but only if the data model can be adapted at runtime (see DP2). Otherwise, always knowing the need in advance will be challenging. It is an increase through better usability of the data, even if the usefulness of the data is not influenced.	Accessibility(+) Importance(∘) Novelty(-) Actability(+) Effectiveness(+)
DP5	The design principle is understood. It is important because the life cycle of data must be considered and managed - and doing this directly in the catalog makes sense. The principle does not provide new insights. That appropriate management of important assets is better than simply storing them somehow is known from other areas. The principle can only be implemented if a life cycle for data has been defined, and this is to be implemented and lived accordingly. Overall, the principle increases the effectiveness. However, the data owner has an additional overhead for the management.	Accessibility(+) Importance(+) Novelty(-) Actability(+) Effectiveness(+)
DP6	The design principle is understood. It is not essential, but it is functional. The principle itself is not new. Many software artifacts with user interaction work with visualizations. It is feasible to create basic visualizations. However, they may only help in some use cases. Sophisticated visualizations may need to be planned and implemented individually for each use case. There is undoubtedly a potential improvement, depending on the quality and flexibility of the visualization.	Accessibility(+) Importance(∘) Novelty(-) Actability(+) Effectiveness(+)

Continued on next page

Table D.3 – continued from previous page

DP7	The design principle is formulated well. It is important because it allows for preserving/ensuring/obtaining/measuring data quality. It has been introduced previously. Furthermore, it is like everywhere else: measuring and tracking the quality of important assets makes sense. Data quality is not a trivial issue to implement (at least if you go beyond "is this CSV complete" now), but review systems and generating alerts are not a problem. Effectiveness is increased. One has an assured possibility to use the highest quality dataset for a use case.	Accessibility(+) Importance(+) Novelty(-) Actability(+) Effectiveness(+)
Overall	The effectiveness of the data catalog depends on much more than just the design principles presented. They certainly create the foundation, but if the search in the catalog does not work well, then even the best automation and visualization will not help. None of the deficiencies mentioned appear to exist.	

D.4 Expert E4

Table D.4: Discussion notes E4

Context	Software Engineer and Architect in Retail Industry E4	Codes
DP1	The term automation may not be appropriate. After all, the design principle's goal is the catalog's completeness. It could be the Principle of Completeness. Automation is just a way to reach completeness. The renaming would make the DP more novel. As it is now, it applies to many systems that collect something in a database. DP1 is appropriate and would assist in selecting or developing data catalogs. DP1 is implementable, although, in detail, it may be more difficult. If the automation is good, it increases the efficiency enormously.	Accessibility(+) Importance(∘) Novelty(∘) Actability(+) Effectiveness(+)
DP2	DP2 is appropriate and would assist in selecting or developing data catalogs. DP2 can be implemented, but the specific details may present more difficulties.	Accessibility(+) Importance(∘) Novelty(∘) Actability(+) Effectiveness(∘)
DP3	This is not new, not even for data catalogs. However, next to automation, it is the most important thing to do to have a functional data catalog. DP3 is appropriate and would assist in selecting or developing data catalogs. While DP3 can be implemented, considering the specific details may be more challenging.	Accessibility(+) Importance(+) Novelty(-) Actability(+) Effectiveness(∘)
DP4	Without context, there is nothing left to justify using a data catalog. DP4 is appropriate and would assist in selecting or developing data catalogs. DP4 is implementable, although in detail, it may be more difficult.	Accessibility(+) Importance(+) Novelty(∘) Actability(+) Effectiveness(∘)
DP5	It has not been thought of yet, but clearly, this is a significant point that is often only implicitly addressed. DP5 is appropriate and would assist in selecting or developing data catalogs. DP5 can be implemented, although it may pose challenges regarding the specifics.	Accessibility(+) Importance(+) Novelty(+) Actability(+) Effectiveness(∘)

Continued on next page

Table D.4 – continued from previous page

DP6	One wants to use visualization to achieve a goal. The Principle of Comprehension/Understanding may better describe the primary goal. Building understanding more easily also means being more efficient. DP6 is appropriate and would assist in selecting or developing data catalogs. It is possible to implement it, although one would also need individual visualizations for different data.	Accessibility(+) Importance(∘) Novelty(∘) Actability(+) Effectiveness(+)
DP7	DP7 is easy to understand. DP7 is appropriate and would assist in selecting or developing data catalogs. The design principle is realistic as long as one sets limits. One cannot evaluate everything and for every use case.	Accessibility(+) Importance(∘) Novelty(∘) Actability(+) Effectiveness(∘)
Overall	One should apply the design principles in practice and see if they have the desired effect. However, since the design principles are derived from practical work, they carry more weight than if they had only been developed theoretically. There is a lack of metrics to measure the success of the design principles. With such abstract design principles, it takes much work to measure. However, on a more concrete level, one needs guardrails to keep one on track. The work already has clear hierarchies with the design principles and subsequent design features. For the next iteration, it would be reasonable to extend it even further for practical use. That is, to develop concrete templates to be reused to design aspects in the data catalog. Nevertheless, one could look at the values the design principles represent and thus possibly build a structure from value to principle to practice. None of the deficiencies could be identified at the design principles level.	

D.5 Expert E5

Table D.5: Discussion notes E5

Context	Data Catalog Consultant in Research E5	Codes
DP1	Chatbots like ChatGPT could help with tasks in the data catalog and automate things. Resources are scarce, so this is important. A data catalog is typically managed by existing employees, not new staff.	Accessibility(∘) Importance(+) Novelty(∘) Actability(∘) Effectiveness(∘)
DP2	Companies have found it helpful to be able to turn off features they do not need based on practical experience. However, this is only possible with a few data catalogs. DP2 and DP3 are pushing against each other. One has to be careful how much flexibility one allows to maintain interoperability. Renaming roles and configuring access rights are some of the most important things to be able to configure individually. The customers want to avoid coding. They want to save time and resources. It has to work out of the box as much as possible. Therefore, it is good if what does not work immediately can be configured via the software.	Accessibility(∘) Importance(+) Novelty(∘) Actability(∘) Effectiveness(+)
DP3	Providing the data catalog as open source makes much sense from a scientific perspective. However, an open-source business model may only be feasible for some data catalog vendors. In practice, this is also extremely important when parts of a company are split off or purchased. One has to be able to bring one's data catalogs together. This is especially important in large companies since they use many different tools that the data catalog must inventory. It would not be feasible if there were no basic interoperability.	Accessibility(∘) Importance(+) Novelty(∘) Actability(∘) Effectiveness(+)

Continued on next page

Table D.5 – continued from previous page

DP4	Data always has a context. To understand it, one needs that context. The data catalog must be able to offer context. This is extremely important. The data catalog should have the functionality to offer the context. Depending on the type of data, one needs to provide different context. However, it is implementable and mandatory. In terms of content, the data catalog can be supported through automation.	Accessibility(∘) Importance(+) Novelty(∘) Actability(+) Effectiveness(∘)
DP5	This is new. This focus has yet to be seen in data catalogs. Diatamini et al. 2018 have a category for operational metadata, but that is it. That would make life easier because one would know immediately if and how to use the data. Management itself is also an important aspect. This way, one has everything in one place.	Accessibility(∘) Importance(+) Novelty(+) Actability(∘) Effectiveness(+)
DP6	It is a very important and understandable feature, especially for non-technical employees. In practice, everyone wanted to visualize data lineage and linkage to make it easier to understand these complex relationships. Even data catalogs that run without an interface can benefit from visualization. At some point, a human must intervene. And there, it would be good to have good visualizations.	Accessibility(+) Importance(+) Novelty(∘) Actability(∘) Effectiveness(+)
DP7	Trust is crucial. Are the data up to date? It is very well supplemented by DP1.	Accessibility(∘) Importance(+) Novelty(∘) Actability(∘) Effectiveness(∘)
Overall	One would have to look somehow at whether the design principles are now complete. This is not easy because they are already on such a high level. The design principles also stand well on their own. They only overlap a little, even though they complement each other well. There are none of the deficiencies mentioned.	

D.6 Expert E6

Table D.6: Discussion notes E6

Context	Computer Scientists and Librarian at a University E6	Codes
DP1	DP1 is understood. Automation of tagging using keywords is an integral part of a library catalog. Three people are just for automating the (meta-)data flow. New kinds of data are not automatically inventoried. A human needs to do the inventory. Pull and Push Strategy used. Push is sometimes an email. Automation is good when using just one software. The most crucial topic for catalogs in a library. Automation would free up many human resources.	Accessibility(+) Importance(+) Novelty(∘) Actability(∘) Effectiveness(+)
DP2	There is no focus on flexibility. However, it saves time sometimes if one can change things to one's needs. Whether the Return on Investment is big enough when one flexibilize is questionable. Flexibilization is expensive. It is clear how one would implement it and also realistic, but it is less critical in cataloging literature currently.	Accessibility(+) Importance(-) Novelty(∘) Actability(+) Effectiveness(+)
DP3	DP3 is understood. Standards are used for keywords. Interoperability is a significant challenge. Two persons are working continuously on the unification of metadata. There is a focus on literature. Without this focus, interoperability is an even bigger problem and must be considered. Standards are used to harvest metadata from different libraries. Metadata formats can suddenly change because a new employee works at the provider. True interoperability would be perfect. Here, too, one would get two whole persons free who could do more important things.	Accessibility(+) Importance(+) Novelty(∘) Actability(∘) Effectiveness(+)

Continued on next page

Table D.6 – continued from previous page

DP4	Much contextual information is stored. For example, whether one needs a new account with the provider, needs to purchase a license, and much more. Network building is a very important emerging point. In the future, networks should flow in from outside and be stacked on each other. This makes it possible to offer excellent search features. Network features are new, and commercial vendors have just started implementing solutions.	Accessibility(∘) Importance(+) Novelty(+) Actability(∘) Effectiveness(+)
DP5	It is imperative to know when licenses expire to remove offers from the catalog. Data is also marked as obsolete and removed if, for example, a newer version exists. Such things are already implemented in other systems but not in the catalog. It would be good to have all this in one place.	Accessibility(∘) Importance(+) Novelty(+) Actability(+) Effectiveness(∘)
DP6	DP6 is understood. Visualizations are essential for professional colleagues. Dashboards are an important element there.	Accessibility(+) Importance(+) Novelty(∘) Actability(∘) Effectiveness(∘)
DP7	DP7 is understood. Assessment of data and metadata happens before insertion into the catalog The topic is currently not that big at libraries for the end user. Applying that to the end-users at the library catalog would be new. Experienced users would have an advantage in being able to assess sources better. They could sort out sources already in the catalog. However, it is generally understandable that it is crucial for data catalogs and helps the users.	Accessibility(+) Importance(+) Novelty(+) Actability(∘) Effectiveness(+)
Overall	After thinking about it, none of the four deficiencies have been identified.	

D.7 Expert E7

Table D.7: Discussion notes E7

Context	Software Engineer and Architect in cloud native environments and for applications in retail supply chains E7	Codes
DP1	DP1 is understood. It is an essential part of data catalogs. One should always have a high degree of automation. Automation should make data in the catalog directly usable to gain new insights based on it. This can be very demanding and should be addressed in companies. Of course, it can be implemented in reality, but as is often the case, it is then neglected. Automation should always be the goal. Otherwise, one can continue to print everything on paper and file it away. It can be implemented. Automation should be on everyone's mind.	Accessibility(+) Importance(+) Novelty(-) Actability(+) Effectiveness(+)
DP2	DP2 is understood. This is nothing new per se. Flexibility can be implemented.	Accessibility(+) Importance(∘) Novelty(-) Actability(+) Effectiveness(∘)
DP3	DP3 is understood. Benefits from automation. Otherwise, it makes little sense. This is very important when many data catalogs need to talk to each other. It saves time. Especially if they are operated by different stakeholders, as is the case with the data space data catalogs. However, this is familiar if one comes from the open source area. Interoperability can be implemented.	Accessibility(+) Importance(+) Novelty(-) Actability(+) Effectiveness(+)
DP4	DP4 is understood. Context is a new concept for many people. If one comes from more of a technical perspective, this term is used relatively rarely. The principle is data catalog-specific. The first three DPs only really become data catalog specific through their design features. The principle is very much use-case specific, but it can be implemented.	Accessibility(+) Importance(∘) Novelty(+) Actability(+) Effectiveness(∘)

Continued on next page

Table D.7 – continued from previous page

DP5	DP5 is understood. This is the unloved stepchild. Everyone knows that it exists, but no one deals with it. That is why mentioning it so explicitly and making people aware of it is good. Operational knowledge about data is crucial; it is a new topic for data catalogs. It is an important topic that runs through many things cross-sectionally. The principle is already being implemented in part but not yet to its full extent. There are still no real solutions that can be applied in reality. However, it is on the way, and something will come of it.	Accessibility(+) Importance(+) Novelty(+) Actability(+) Effectiveness(∘)
DP6	DP6 is understood. Whenever a user is supposed to interact with a system, good visualization pays off. It is thus an important design principle. With complex systems and data, even more so. This is a non-trivial topic. If one does it well, the application shows, e.g., a graphical tree at the end that one understands immediately. This makes the work more efficient. Sometimes, people ask why it took so many years to implement it. However, that is precisely the effect one wants. Often, one comes from the visualization because the customer first considers it that way.	Accessibility(+) Importance(+) Novelty(∘) Actability(∘) Effectiveness(+)
DP7	DP7 is understood. It is important when thinking about a data marketplace. The sample is an essential component here. When one has to make decisions, one needs this tool. It could be a kind of data purchasing advisor. DP7 is undoubtedly the most interesting for research. One could ship one's own AIs or algorithms to generate interesting metrics most relevant to oneself.	Accessibility(+) Importance(+) Novelty(∘) Actability(+) Effectiveness(∘)

Continued on next page

Table D.7 – continued from previous page

Overall	All of the principles are useful. DP1 and DP3 complement each other well. They focus on different things, but both would be good for data catalogs. The DPs are partly lived practice, but one must remember them. And that is what this listing is good for. The seven design principles cover a vast field. They certainly cover the whole field of data catalogs here, if not other types of systems that are similar. Topics like data governance and audits are not mentioned but are inside DP5. Scalability would be in DP2. Specifically, there is not anything that is missing. Nothing should be summarized. Otherwise, everything becomes way too abstract. There may be slight overlaps, but there are no gaps this way. No design principle can be omitted. Most likely automation, but one does not want that. It is possible but not clever.

D.8 Expert E8

Table D.8: Discussion notes E8

Context	Data Expert for Cities and Municipalities E8	Codes
DP1	The design principle is formulated in an accessible way. Automation is very relevant because one often wants to have data from people who do not work with data daily. Moreover, in the smart city area, one only wants to have a one-time effort, and then it has to run. The principle does not give any novel insights, but in the context of data catalogs, it gives new insights in interaction with all the other principles. Automating profiling is important. No one is going to document authors on an Excel file. Automation is feasible in a data catalog, but it is costly. This principle certainly increases efficiency and productivity the most.	Accessibility(+) Importance(+) Novelty(+) Actability(+) Effectiveness(+)
DP2	The design principle is formulated in an accessible way. Flexibility is a must in the smart city environment. Often, particular fields are needed to store historically grown internal numbers and the like. The principle does not give any novel insights, but in the context of data catalogs, it gives new insights in interaction with all the other principles. It would greatly simplify the work with different cities, and one can implement the needs much faster.	Accessibility(+) Importance(+) Novelty(+) Actability(∘) Effectiveness(+)
DP3	The design principle is formulated in an accessible way. The design principle is technically important, and it is necessary for data catalogs to communicate. However, it is not currently asked for by customers. While the principle itself does not offer any unique insights, it provides new perspectives when applied within the context of data catalogs. Interoperability is doable if one sticks to a few standards. It already helps if one uses DCAT in the data catalog for harvesting. It gets more complicated when pulling data from a foreign system for the first time. Then, one has to connect systems step by step gradually. This principle certainly increases efficiency and productivity the most.	Accessibility(+) Importance(+) Novelty(+) Actability(+) Effectiveness(+)

Continued on next page

Table D.8 – continued from previous page

DP4	The design principle is formulated in an accessible way and is important for data catalogs. The principle does not provide any new insights on a high level, but the data network is new to ODPs. The principle is technically feasible but requires more work to implement in practice. One should collect as much context as possible, but it fails because of many domains and specializations. A considerable degree of automation would have to be achieved, but that would involve time and effort. Likewise, entering by hand is also an effort; one can only expect people to do that up to a certain amount of fields. It makes working with data much more manageable for cities because they need their various internal numbers. The principle increases the effectiveness.	Accessibility(+) Importance(+) Novelty(+) Actability(-) Effectiveness(+)
DP5	The design principle is formulated in an accessible way. The design principle is important and necessary for data catalogs. From a technical standpoint, the need will arise sooner than from the customer's perspective. At least for ODPs. However, it has already happened that data suddenly has to be deleted from the data catalogs because the originator of the data has changed his license. Comprehensive life cycle management would benefit both the data and the metadata. Then, one could clarify many things without having to make phone calls. There are also use cases where data is only issued for a short time. It would be good to be able to say that the data will expire after 12 months and that everyone knows this. The design principle is novel in the ODP context.	Accessibility(+) Importance(+) Novelty(+) Actability(∘) Effectiveness(+)

Continued on next page

Table D.8 – continued from previous page

DP6	The design principle is formulated in an accessible way. From the perspective of the end user, the principle is fundamental. First and foremost, it is a good selling point. It also helps the typical end user gain insight and increase productivity. The professionals certainly use their own visualization tools with the data. Many data portals don't have good visualizations because they often only make sense if you manually adjust them to the data. The design principle is novel precisely because visualization is not yet widely practiced in the ODP field. There is still much potential.	Accessibility(+) Importance(+) Novelty(+) Actability(∘) Effectiveness(+)
DP7	The design principle is formulated in an accessible way. The design principle is important in the implementation of data catalogs. The principle would be new to the ODP world, as few assessments are made. Often, there is not even a rating system. Discussions with technically savvy people often reveal that such an assessment is challenging. It can be implemented within a particular scope. In detail, one must then look at what the term quality means in a given context. Such an assessment increases efficiency in working with data. One quickly knows whether one has to correct the data again or whether one should use it.	Accessibility(+) Importance(+) Novelty(+) Actability(+) Effectiveness(+)

Continued on next page

Table D.8 – continued from previous page

Overall	Design principles are understandable for the target group. For those who work in the cities, it would be partially unclear. The list of design principles is new, and it does not exist yet. Even if one is familiar with the principles, they help to keep an eye on them. One can go to customers with the principles and introduce them as cornerstones. And then discuss where one wants to put more energy into. The Principle of Automation and the Principle of Flexibility push against each other. One can not have 100% automation and 100% flexibility. Automation and interoperability play very well together. With automation, one also often standardizes. There could also be some redundancy. Not technically, but in terms of the goal. Both aim to have a complete data catalog. The visualization may be an excess for some people because they use their own visualization tools. However, in general, it already makes sense to address visualization. Especially to see the graph with the links is already helpful to keep the overview.	Principle Redundancy Principle Excess

E Own Publications

The following appendix contains a list of publications in which the author of the dissertation was involved. First, the publications relevant to this thesis are listed, and the author's contribution is described. Publications are considered relevant if they report on design knowledge that can be used to implement data catalogs. Publications were also considered relevant if they were part of a research series leading to a paper containing design knowledge for data catalogs. Afterward, publications are listed in which the author was involved but are not relevant in the context of this thesis.

Relevant Publications for This Dissertation

I. Spiekermann, Markus, **Daniel Tebernum**, Sven Wenzel, and Boris Otto: 'A metadata model for data goods'. *Proceedings of the Multikonferenz Wirtschaftsinformatik (MKWI) 2018*. Ed. by Drews, Paul, Burkhardt Funk, Peter Niemeyer, and Lin Xie. Vol. 2018. ISBN 978-3-935786-72-0. https://www.leuphana.de/fileadmin/user_upload/Forschungseinrichtungen/iis/files/MKWI2018/MKWI2018_Band1.pdf [Accessed: August 16, 2022]. 2018: pp. 326–337.

II. **Tebernum, Daniel**, Markus Spiekermann, Sven Wenzel, and Boris Otto: 'Risikobewertungen in Datennetzwerken'. *D · A · CH Security 2018 : Bestandsaufnahme, Konzepte, Anwendungen, Perspektiven*. Ed. by Schartner, Peter and Norbert Pohlmann. ISBN 978-3-00-060424-9. https://www.syssec.at/de/veranstaltungen/dachsecurity2018/papers/DACH_Security_2018_Paper_23A1.pdf [Accessed: August 16, 2022]. Frechen : syssec, 2018: pp. 287–297.

III. **Tebernum, Daniel** and Dustin Chabrowski: 'A Conceptual Framework for a Flexible Data Analytics Network'. *Proceedings of the 9th International Conference on Data Science, Technology and Applications, DATA 2020, Lieusaint, Paris, France, July 7-9, 2020*. Ed. by Hammoudi, Slimane, Christoph Quix, and Jorge Bernardino. DOI https://doi.org/10.5220/0009827402230233. https://www.scitepress.org/PublishedPapers/2020/98274/98274.pdf [Accessed: August 16, 2022]. SciTePress, 2020: pp. 223–233.

IV. **Tebernum, Daniel**, Marcel Altendeitering, and Falk Howar: 'DERM: A Reference Model for Data Engineering'. *Proceedings of the 10th International Conference on Data Science, Technology and Applications, DATA 2021, Online Streaming, July 6-8, 2021*. Ed. by Quix, Christoph, Slimane Hammoudi, and Wil M. P. van der Aalst. DOI https://doi.org/10.5220/0010517301650175. https://www.scitepress.org/PublishedPapers/2021/105173/105173.pdf [Accessed: August 16, 2022]. SCITEPRESS, 2021: pp. 165–175.

V. **Tebernum, Daniel**, Marcel Altendeitering, and Falk Howar: 'A Survey-Based Evaluation of the Data Engineering Maturity in Practice'. *Data Management Technologies and Applications*. Ed. by Cuzzocrea, Alfredo, Oleg Gusikhin, Slimane Hammoudi, and Christoph Quix. DOI https://doi.org/10.1007/978-3-031-37890-4_1. https://link.springer.com/chapter/10.1007/978-3-031-37890-4_1 [Accessed: August 2, 2023]. Cham: Springer Nature Switzerland, 2023: pp. 1–23.

VI. **Tebernum, Daniel** and Falk Howar: 'Structuring the End of the Data Life Cycle'. *Proceedings of the 12th International Conference on Data Science, Technology and Applications - DATA*. DOI https://doi.org/10.5220/0011999300003541. https://www.scitepress.org/Papers/2023/119993/119993.pdf [Accessed: August 2, 2023]. INSTICC. SciTePress, 2023: pp. 207–218.

VII. **Tebernum, Daniel** and Falk Howar: 'Treating the End of the Data Life Cycle as a First-Class Citizen in Data Engineering'. *Wirtschaftsinformatik 2023 Proceedings. 8*. https://aisel.aisnet.org/wi2023/8. https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1007&context=wi2023 [Accessed: Oktober 4, 2023]. 2023.

Comments on My Participation

- **A Metadata Model for Data Goods**
  The data model was developed equally by the first author and me. I led and carried out the necessary development work for the evaluation to integrate the data model into DIVA. I contributed to all chapters.
- **Risikobewertung in Datennetzwerken**
  I am the primary author of this paper. I mainly developed chapters 2, 3, and 4. Chapters 1 and 6 were written equally by me and the second author. I supported the development of chapter 5. I led and carried out the technical ideation and implementation in DIVA.
- **A Conceptual Framework for a Flexible Data Analytics Network**
  I am the main author of this paper. I mainly wrote chapters 1 and 2. The authors wrote chapters 3, 4, 7, and 8 equally. I conceived the underlying concept of the paper. The technical implementation by the co-author was supervised by me.
- **DERM: A Reference Model for Data Engineering**
  The research results of this publication were collaboratively developed by the authors. The development of the paper was led by me. I am the main author responsible for chapter 3, chapter 4, the formulation of research questions, and the visualization of the reference model. I supported the development of the remaining chapters.
- **A Survey-based Evaluation of the Data Engineering Maturity in Practice**
  I am the main author of this paper and have supervised all chapters. The underlying survey was developed through collaboration among all authors. I led the data analysis and visualization and implemented them in Python. In chapter 4, in particular, the analyses on the Data Engineering phases of *Plan*, *Select/Access*, and *Destroy* were created by me.

- **Structuring the End of the Data Life Cycle**
  I am the main author of this paper and have written all chapters. The underlying idea was identified by me. I created all analyses, chapters, and visualizations.
- **Treating the End of the Data Life Cycle as a First-Class Citizen in Data Engineering**
  I am the main author of this paper and have written all chapters. The underlying idea was identified by me. I created all analyses, chapters, and visualizations.

Other Peer-Reviewed Publications

1. Pettenpohl, Heinrich, **Daniel Tebernum**, and Boris Otto: 'WFDU-net: A Workflow Notation for Sovereign Data Exchange'. *Proceedings of the 10th International Conference on Data Science, Technology and Applications, DATA 2021, Online Streaming, July 6-8, 2021*. Ed. by Quix, Christoph, Slimane Hammoudi, and Wil M. P. van der Aalst. DOI `https://doi.org/10.5220/0010550402310240.https://www.scitepress.org/PublishedPapers/2021/105504/105504.pdf` [Accessed: August 16, 2022]. SCITEPRESS, 2021: pp. 231–240.
2. Steinert, Michael, **Daniel Tebernum**, and Marius Hupperz: 'Design Features for Data Trustee Selection in Data Spaces'. *Proceedings of the 13th International Conference on Data Science, Technology and Applications - DATA*. DOI `doi.org/10.5220/0012851400003756. https://www.scitepress.org/Papers/2024/128514/128514.pdf` [Accessed: July 16, 2024]. INSTICC. SciTePress, 2024: pp. 559–570.

Acknowledgments

I would like to express my deepest gratitude to my family. To my wife, Sarah, my daughter, Sophie, and my son, Jonas, thank you for your unwavering support, patience, and love throughout this journey. You have been my pillars of strength. I also extend my heartfelt thanks to my parents for their constant encouragement and belief in me.

I am profoundly grateful to my supervisor, Prof. Dr. Falk Howar, for granting me the freedom to explore my ideas while providing crucial feedback and guidance. Your support has been invaluable. My sincere thanks also go to my second supervisor, Prof. Dr.-Ing. Frederik Möller, for imparting essential knowledge about Design Science Research, significantly enriching this work.

To my colleagues and former colleagues, thank you for your support and camaraderie. I would like to particularly acknowledge my co-authors and DIVA engineers—Marcel Altendeitering, Sergej Atamantschuk, Dustin Chabrowski, and Markus Spiekermann—for their invaluable collaboration and contributions to this work. A special thanks to Julia Pampus and Dr.-Ing. Fabian Bruckner for our numerous engaging discussions about my research. Your insights and friendship have been deeply appreciated.

Thank you all for believing in me.