TU WIEN Informatics

Automatisierte Qualitätsindikatoren für Maschinell Verarbeitbare Datenmanagementpläne

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering & Internet Computing

eingereicht von

Lukas Arnhold, BSc.
Matrikelnummer 11709460

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Dr.techn. Mag. Tomasz Miksa
Mitwirkung: Dipl.-Ing. Moritz Staudinger

Wien, 31. Juli 2024

Lukas Arnhold　　Tomasz Miksa

Informatics

Automated Quality Indicators for Machine-actionable Data Management Plans

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering & Internet Computing

by

Lukas Arnhold, BSc.
Registration Number 11709460

to the Faculty of Informatics

at the TU Wien

Advisor: Dr.techn. Mag. Tomasz Miksa
Assistance: Dipl.-Ing. Moritz Staudinger

Vienna, July 31, 2024

Lukas Arnhold Tomasz Miksa

Erklärung zur Verfassung der Arbeit

Lukas Arnhold, BSc.

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 31. Juli 2024

Lukas Arnhold

Acknowledgements

First of all, I would like to thank my wonderful parents for their continuous and unconditional support.

In addition, I want to thank Gabi who has always had an open ear for my problems and endured and supported me through the challenging times of writing this thesis.

Special thanks to my supervisors Tomasz Miksa for guiding me through this thesis and the opportunity to work on the OSTrails project, and Moritz Staudinger for your invaluable feedback and support.

Kurzfassung

Datenmanagementpläne (DMPs) sind Dokumente die Forschungsprojekte begleiten und beschreiben, wie Daten verwaltet, persistiert und wiederverwendbar gemacht werden. Maschinell verarbeitbare DMPs (maDMPs) beinhalten strukturierte Metadaten auf denen Anwendungen aufbauen können und die Research Data Alliance (RDA) veröffentlichte den DMP Common Standard (DCS) als Referenz. Lösungen, die maDMPs verwenden, decken bereits verschiedene Anwendungsfälle ab, doch trotz dieser Fortschritte wird die Überprüfung von DMPs in der Regel manuell durchgeführt, und unseres Wissens existiert keine automatisierte Lösung, die Reviewer dabei unterstützt. Folglich ist die Evaluierung von DMPs ein zeitintensiver, manueller Prozess, der stark vom Urteil der Reviewer abhängig ist.

In dieser Arbeit untersuchen wir die Verwendung automatisierter Indikatoren, um Informationen zu Qualitätsaspekten bereitzustellen, um Reviewern bei der Evaluierung von DMPs zu unterstützen. Basierend auf einem Review der Literatur und Diskussionen mit der maDMP Community identifizieren wir Anforderungen und schlagen eine konzeptionelle Architektur für ein Framework vor, welches Reviewern bei der Evaluierung von DMPs durch die Bereitstellung von automatisch generierten Qualitätsindikatoren unterstützt.

Aufbauend auf dieser konzeptionellen Architektur implementieren wir einen Prototypen und evaluieren diesen, um zu überprüfen, ob die ermittelten Anforderungen abgedeckt sind und zeigen durch konkrete Beispiele, dass der Prototyp automatisch Qualitätsmessungen bereitstellen kann, um Evaluierungskriterien in Bezug auf Vollständigkeit, Durchführbarkeit, Qualität der Aktionen und Richtlinienkonformität abzudecken und, dass diese Messungen das Ergebnis einer manuellen Evaluierung reflektieren.

Außerdem führen wir eine Fallstudie durch, um herauszufinden, inwieweit die DMP-Evaluierung in Bezug auf die Anforderungen der Science Europe Evaluation Rubric mithilfe der vorgeschlagenen Lösung automatisiert werden kann. Das unveränderte DCS-Anwendungsprofil kann nur bedingt die Informationen, die in der Evaluation Rubric verlangt werden, abdecken. Nach der Erweiterung des DCS-Anwendungsprofiles können wir eine Zuordnung erstellen, die alle von der Evaluation Rubric benötigten Information zu Elementen eines maDMP zuordnet. Allerdings stellen wir auch fest, dass nur eine teilweise automatisierte Evaluierung möglich ist und genauere Bewertungskriterien erforderlich sind, um einen höheren Automatisierungsgrad zu erreichen.

Abstract

Data Management Plans (DMPs) are documents accompanying research projects with the purpose of describing how data is managed, stored and preserved. Machine actionable DMPs include structured metadata that applications can leverage and the Research Data Alliance (RDA) proposed the DMP Common Standard (DCS) as a baseline. Applications using maDMPs already address various use cases, but nonetheless the review of DMPs is usually conducted manually by human reviewers, and to the best of our knowledge, there exists no solution to assist them. Therefore, the evaluation of DMPs is a time intensive process involving extensive human labor and judgement.

In this work, we investigate the use of automated indicators to provide information on quality aspects to assist reviewers in the evaluation of DMPs. Based on the literature review and discussions with the maDMP community we identify requirements and propose a conceptual architecture of a framework assisting in the evaluation of DMPs through the provision of automatically created quality indicators.

Following the conceptual architecture, we implement a prototype and evaluate it to verify that the elicited requirements are covered and show through concrete examples, that the prototype can automatically provide quality measurements to meet the review goals completeness, feasibility, quality of actions and guideline compliance and that these measurements mirror the results of a manual assessment.

In addition, we perform a case study to show to what extent the DMP evaluation with regard to the Science Europe DMP Evaluation Rubric can be automated using the proposed solution. Using the unaltered DCS application profile only parts of the guidelines in the Evaluation Rubric can be addressed with corresponding information of a DCS maDMP. After extending the underlying DCS application profile we can provide mappings covering the required information of all given questions of the rubric, but we also observe that only partial automated assessment is possible and more machine actionable evaluation criteria are needed to achieve a higher level of automation.

Contents

CHAPTER 1

Introduction

Data Management Plans (DMP) are documents accompanying research projects during the different stages of the research lifecycle with the intention of describing the data used and produced during research activities, as well as illustrating how data will be handled according to FAIR principles. The FAIR principles, an acronym for Findable, Accessible, Interoperable and Reusable, have been proposed by Wilkinson et al. [WDA+16] and are a set of guiding principles to help promote good data management practices in scientific and research contexts. As a tool that raises awareness of good data management, various research funding bodies such as the National Science Foundation (NSF) in the United States and the European Commission in Europe require researchers to submit a DMP when applying for research grants [MSMJ19, KM15, MOR22].

Although DMPs are already widely required by funding bodies, their review is in most cases done manually, with reviewers reading a text document prepared by the researcher according to the DMP template of the funder [MSS+23]. Due to the heterogeneity of research practices across disciplines and the use of DMPs as a tool to collect information from a variety of resources, the quality of the feedback depends on the experience of the reviewers and can vary extensively.

Originally, DMPs used to be text documents that were not machine readable, making the review process not only a labor intensive manual task, but also limiting their utility to the stakeholders involved, so that they have often been viewed as an administrative burden [MSMJ19]. Therefore, the Research Data Alliance (RDA) proposed the DMP Common Standard (DCS) for Machine Actionable Data Management Plan (maDMP)s to improve the value of DMPs for stakeholders involved in Research Data Management (RDM), by allowing them to exchange information using this common standard [MWN+21, MWN20]. As a result, certain aspects of Research Data Management (RDM) could already be automated using maDMPs, thus reducing the administrative burden and improving RDM processes [WO20, Zou23, SJMM17]. Furthermore, the development of the DMP Common Standard Ontology (DCSO) serialization provides a representation of maDMPs using a controlled RDF vocabulary that enables the use of semantic web technologies in the context of maDMPs.

1.1 Motivation

Although the evaluation and compliance monitoring of the content contained in maDMPs is seen as an important part of the effort to integrate maDMPs into the broader RDM infrastructure [MSMJ19, SJMM17, PMSBG23, JPH+20], the review of DMPs is carried out manually in most cases, and to our knowledge, there exists no integrated approach to enable automatic or semi automated evaluation of DMPs. Miksa et al. [MSS+23] compiled an overview of recent case studies, shown in Table 1.1, with the objective of presenting methods on how maDMPs could be automatically evaluated. Although these tools show the possibility of automatic evaluation, they only aim to show the feasibility of specific aspects and do not constitute a universal solution regarding the evaluation of DMPs.

However, in practice, reviewers usually manually evaluate textual DMPs based on evaluation criteria such as the Science Europe Evaluation Rubric [Eur21] and must manually gather the contextual information required from resources such as Scientific Knowledge Graph (SKG). Manual evaluation requires intricate domain knowledge and expertise that strongly influences the quality of the evaluation. Especially since a wide variety of aspects, such as metadata standards, repositories, licensing, and accordance to FAIR principles, among others, have to be taken into account.

Miksa et al. [MSS+23] investigated different scenarios of usage and requirements of such an automatic maDMP evaluation solution. Figure 1.1 depicts the current state of the art (*AS-IS*) and presents two possible solutions on how to automate the review process (*TO-BE-1* and *TO-BE-2*). In *AS-IS*, although the DMP might already be available as an maDMP it is converted into a text-based document that answers questions of the funder's template before sending it to the reviewer. Therefore, in the current review process, the benefits of maDMPs cannot be exploited, leading to significant work overhead and feedback that is highly dependent on the reviewer's expertise.

In *TO-BE-1* the researchers use their current workflows to create a maDMP and send it to the funder who has dedicated evaluation software. The software provides structured and human-readable information and metrics to help reviewers in their evaluation and reduce the amount of manual research needed to provide contextual information. If the

[1] https://github.com/fekaputra/MadPot
[2] https://github.com/raffaelfoidl/maDMP-evaluation
[3] https://github.com/RDA-DMP-Common/RDA-DMP-Common-Standard/tree/master/ontologies/validation
[4] https://github.com/vknaisl/madmp-metadata-standard-link-evaluation
[5] https://fairsharing.org/
[6] https://ds-wizard.org/
[7] https://github.com/MarekSuchanek/madmp-fairness-evaluation
[8] https://www.f-uji.net/

Existing Solution	Description
MadPod[1]	Provides transformation between maDMP JSON serialization and validation of the provided SHACL shapes. A set of SHACL shapes to validate parts of a DCS DMP are provided.
Funder Requirement Evaluation[2]	Foidl et al. [FBM21] Provide SPARQL queries to query information required to answer the Science Europe DMP guidelines.
DCSO ShEx Validation[3]	The repository containing the official publication of the DCS ontology also contains SHEX files to validate aspects of a DCS DMP.
Check Metadata Links[4]	Verfies if metadat standards provides in a maDMP are registered in a trusted source, in this case FAIRSharing[5]
Data Stewardship Wizard[6]	The Data Stewardship Wizard is a tool that helps to create DMPs using questionnaires. In addition to the creation of DMPs according to a given template, it also offers a variety of functions in the context of DMP creation. Regarding evaluation it provides the possibility to give FAIR guidance to answers given in the questionnaire.
F-UJI Evaluator[7]	This tool extracts all identifiers of all datasets in a maDMP and retrieves the achieved FAIRness of the hosted dataset using the F-UJI Evaluator[8].

Table 1.1: Case studies on aspects of automatic evaluation of maDMPs [MSS+23].

funding agency does not intend to use an evaluation software, researchers can still benefit from an automatic evaluation of the DMP before submitting the document as shown in *TO-BE-2*.

In summary, there currently exists no unified architecture that leverages maDMPs to automatically evaluate DMPs and assist reviewers in their work. However, the ability to do so has already been recognized in publications that elaborate on the requirements of machine-actionable DMPs as an important contribution [JPH+20, MSMJ19, MRGB17, CCE+22, SJMM17]. Based on previous work by Miksa [MSS+23], current FAIR practices [WDA+16] and DMP evaluation guidelines [Eur21, WCW+16, BUD+19] we identified the following review goals that will be addressed in this thesis:

- **G1. Completeness**: Evaluate whether all relevant aspects of a maDMP are covered. The DCS provides guidance on the required fields in a valid maDMP. This standard can be used to validate a given maDMP against and highlight sections that do not conform to the standard, are missing, or are only partially available. As such, for example, the license for a dataset could be missing, or more generally there could exist entities that have an incorrect format or are available in incorrect multiplicities.

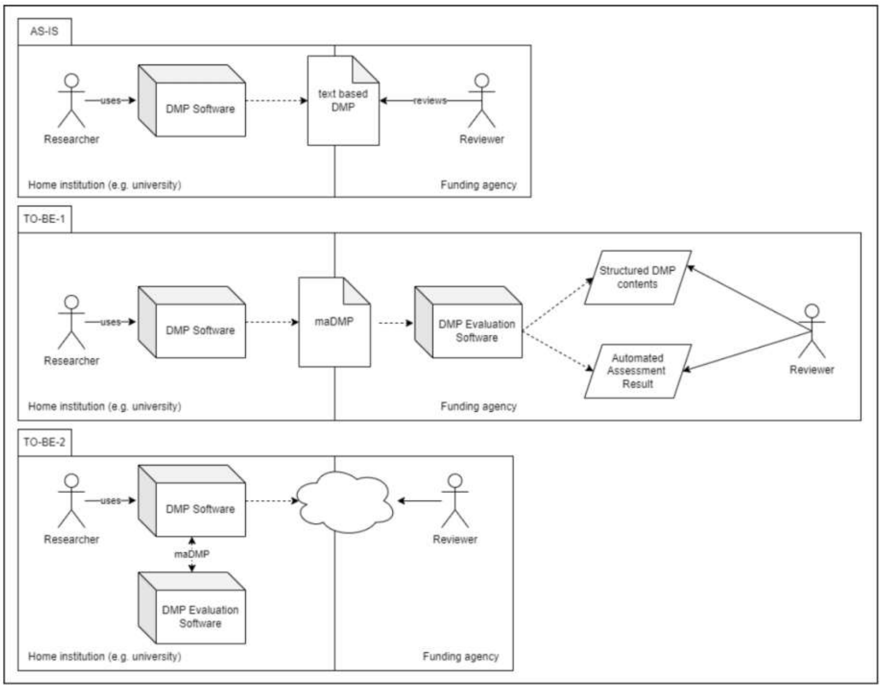

Figure 1.1: Current DMP evaluation scenario (AS-IS) and possible scenarios for automation on maDMP evaluation (TO-BE) [MSS+23].

- **G2. Feasibility**: Evaluate the possibility that the content of a maDMP can be put into practice through concrete actions, which depends on the current stage of the lifecycle of the DMP, particularly if the actions are just described or already implemented:
  - Evaluate whether there is inconsistent or contradictory information in a given maDMP. For example, a funder has an open access policy, but the published dataset has a closed license.
  - Evaluate whether the actions described in a DMP are followed through. Funders, institutions, and repository managers would benefit from an automated system to verify that researchers have met the commitments they made in their DMPs. Simms et al. [SJMM17] defined this as an essential requirement for maDMPs.
- **G3. Quality of Actions:** Evaluate the relevance and effectiveness of the actions listed in the maDMP depending on the data lifecycle phase.
  - Evaluate if the actions described in the DMP lead to FAIR data. In particular, Miksa et al. [MRGB17] mentioned the need to evaluate reproducibility and proposed creating new metrics.

  - Evaluate the FAIRness that has been achieved. The FAIR community provides metrics and standards to assess whether digital objects comply with community standards. These standards can be integrated in the context of a maDMP evaluation solution to provide guidance on the FAIRness achieved at later stages of the data lifecycle.

- **G4. Guideline Compliance**: Evaluate whether a given maDMP complies with DMP guidelines. For example, some funding bodies require DMPs to comply with their guidelines, but in addition to funder requirements there exist other guidelines that are of importance such as the DCS standard itself to guarantee the proper form of a given maDMP.

1.2 Goal and Scope

The overarching goal of this thesis is to design a framework that will aid different stakeholders involved in a project's DMP in the review process by providing indicators based on the data provided by a DMP. Indicators should collect and refine information to help reviewers get a clearer understanding of the DMP at hand. As a consequence, a constraint of the given solution is that it should not provide an assessment of the given DMP but rather augment the human-based assessment with the indicators provided. These should also be explainable and unbiased, and therefore techniques like free-text analysis will not be considered, but listed as future work due to the additional challenges involved.

More specifically, this thesis aims to further develop ideas on automation of the evaluation of maDMPs proposed by the community and to show their feasibility. The solution should assist DMP reviewers at different stages of the research-data lifecycle, such as when datasets are fist described or after they have been persisted into a repository. The solution should integrate information from different sources into the evaluation and provide information and indicators to support reviewers.

The following components are expected to be delivered as part of the thesis:

- Taxonomy of goals G1 to G4 into more fine-grained dimensions and corresponding indicators to describe the results of the automated evaluation of DMPs.
- Proposal of a mapping of information included in a DCS DMP to entities of SKGs and repositories.
- Methods to produce measurements for goals G1 to G4. In addition to a proposal of a decomposition of these goals, the thesis should also propose methods to produce measurements for the corresponding indicators.
- Conceptual design of an evaluation framework for the automatic provision of indicators for a given DMP on the goals G1 to G4 to help reviewers with the evaluation of DMPs in different stages of the data life cycle.

- Extension of the DCS standard to enable better coverage of selected funder's DMP requirements.

1.3 Research Questions

Based on the motivation described in Section 1.1 and the description of the expected results described in Section 1.2, following research questions should be answered in this thesis:

- **RQ1**: To what extent can maDMPs be automatically validated with respect to goals G1 - G4?
  - What are suitable indicators to measure goals G1 - G4 in the context of DMPs and how are they embedded in existing standards?
  - What is the correct way to combine semantic web technologies, SKGs, and FAIR evaluation tools to provide measurements of the proposed indicators and to what extent can automation be achieved?
- **RQ2**: To what extent can the coverage of funder guidelines according to G4 Guideline Compliance be increased by adapting the RDA DMP Common Standard?
- **RQ3**: What is a conceptual framework that provides automated indicators that measure aspects of the quality of a maDMP at different stages of the research data lifecycle?

1.4 Methodology

The methodological approach of this thesis follows the principles of Design Science of Hevner et al. [HMPR04, Hev07]. The design science paradigm is fundamental to the discipline of information systems, which aims to create new and innovative artifacts to solve identified organizational problems. Figure 1.2 shows the three cycles defined in the design science paradigm (*Design Cycle*, *Relevance Cycle*, *Rigor Cycle*) adapted for the needs of this thesis. During the *Relevance Cycle* opportunities and challenges in the application environment will be researched. This cycle is supposed to provide requirements and define criteria for the evaluation of the solution. The *Rigor Cycle* grounds the research project by providing past knowledge and experience while also adding new knowledge, created during the research, back to the Knowledge Base (KB). Lastly, the *Design Cycle* represents the central cycle of design science research projects and iterates between the construction and evaluation of an artifact to further refine it. Due to the scope of the thesis, only one cycle will be completed.

Based on these principles of design science research, the following steps will be carried out in this thesis:

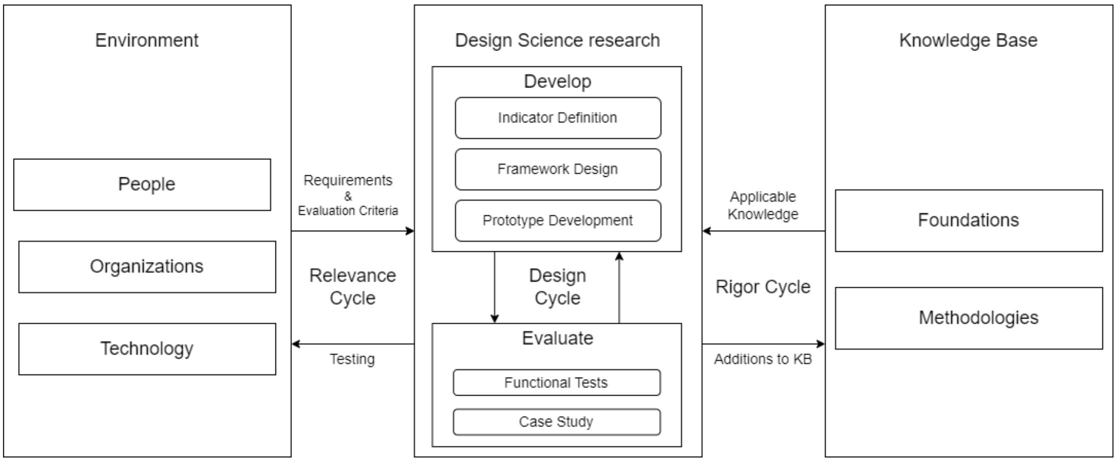

Figure 1.2: Overview of the Methodology of this thesis in the context of the cycles defined by the principles of Design Science [HMPR04]. Adapted and extended from Hevner [Hev07].

1. **Identify relevant problems**: The current state-of-the-art in DMP evaluation will be investigated and possibilities for automated evaluation using maDMPs will be identified. This includes further investigation of the goals and corresponding methods involved in the evaluation. The benefits of a solution that enables the automated evaluation of maDMP will be highlighted. This corresponds to the Design-Science *Relevance Cycle*. At the end of this step, the requirements for the solution, as well as evaluation test cases and scenarios, will be defined.

2. **Design**: Artifacts will be designed for an automated maDMP evaluation solution to address the requirements identified in the *Relevance Cycle*. This step corresponds to the *Design Cycle* and is therefore grounded in existing knowledge through the *Rigor Cycle*, meaning that methodologies from the knowledge base of design science will be applied. To this extent, DMP quality indicators will be defined and an evaluation framework will be designed using common system engineering methodologies. Artifacts involving semantic web technologies will be modeled using standards approved by the World Wide Web Consortium (W3C) where possible.

3. **Evaluation**: As part of the *Design Cycle*, the proposed architecture will be instantiated as a prototype and evaluated using the evaluation criteria defined during the *Relevance Cycle*. For the evaluation of the solution, only functional requirements will be taken into account, and non-functional ones such as performance will not be considered in this thesis. To evaluate the overall approach, the evaluation rubric published by Science Europe [Eur21] will be taken as a baseline to represent the compliance requirements of funders in a case study.

4. **Share Results**: A solution for the given problem will be proposed and shared with the RDA Active DMP interest group. The solution will be discussed and

shortcomings, as well as possibilities for future work, will be identified.

1.5 Thesis Structure

The thesis is structured into the following chapters:

- **Chapter 2 - Related Work**: Introduces related work and provides a background for the contributions of this thesis. In this chapter, we cover topics related to the FAIR principles, implementation of these principles, implementation of corresponding evaluators, and the topic of FAIR enabling resources. Furthermore, we cover data quality dimensions found in literature and discuss associated developments, including standards to express those measurements. We provide an overview of semantic web technologies and their applications, as well as KG and SKGs. Furthermore, we cover topics related to recent developments of maDMPs and how DMPs fit into research data management workflows. We highlight the necessity of maDMPs and present the DCS standard proposed by the RDA as well as possible serializations and proposed extensions of this standard.

- **Chapter 3 - Requirements**: This chapter elaborates on the requirements of a solution that provides indicators to measure quality aspects of maDMPs, by providing use cases, functional requirements, as well as non-functional requirements.

- **Chapter 4 - Conceptual Design**: In this chapter we propose a framework for the evaluation of maDMPs that covers the elaborated requirements for the evaluation of DMPs as well as a proposal on how to provide measurements for the given evaluation dimensions. To communicate the proposed solution, we build upon tested methods to propose a comprehensive Enterprise Architecture (EA) describing both the business as well as the information systems architecture.

- **Chapter 5 - Implementation**: In this chapter, we provide information on the implementation of a prototype of the proposed reference architecture that covers the proposed requirements. To provide a basis for the further evaluation of the utility of the proposed solution, we implemented a case study for the automated evaluation of funder requirements based on the Science Europe DMP evaluation rubric and the evaluation dimensions proposed in this thesis.

- **Chapter 6 - Evaluation**: In this chapter, we evaluate the implemented prototype to verify whether the solution fulfills the given requirements. In addition to showing the efficacy of the prototype we provide illustrative examples of quality measurements produced by the solution and evaluate the extent to which the DMP evaluation goals of this thesis are covered. Furthermore, we evaluate the coverage of DMP funder requirements and what level of automation of the Science Europe evaluation rubric can be achieved.

- **Chapter 7 - Conclusion and Future Work**: In the last section, we draw a conclusion and discuss the outcomes of this thesis. Lastly, we point to open questions regarding the field of automated evaluation of maDMPs and outline possibilities of future work in this field.

CHAPTER 2

Related Work

2.1 FAIR Data

The FAIR principles proposed by Wilkinson [WDA+16] describe quality standards for digital objects with respect to findability, accessibility, interoperability, and reusability. Since the original publication of the guiding principles, these concepts have been refined into maturity indicators [Gro20] that can be used as a checklist for manual evaluation of FAIRness as well as for automatic evaluation of digital objects using tools such as F-UJI [DH20], FAIR-Checker [GRDL+23] or FAIR Evaluation Services [WDS+19]. In the scope of DMPs, the Data Stewardship Wizard [PHS+19] provides a questionnaire that maps the responses to the achieved FAIRness. In addition, tools such as FairSharing help identify standards, databases, or repositories for certain disciplines [SMRS+19].

Since the publication of high-level FAIR principles by Wilkinson et al. [WDA+16] various derivatives emerged, proposing different interpretations of these principles to fit the needs of different communities, but can also lead to different evaluation results for the same digital object. Moser et al. [MWH+23] summarize the developments of important FAIR standards since the publication of the FAIR principles and provide a simplified overview depicted in Figure 2.1. It depicts the FAIR Maturity Indicators MI by Wilkinson et al. [WDS+19], the RDA FAIR Data Maturity Model, the FAIRsFAIR Data Object Assesment Metrics, and the FAIR Metrics for EOSC as important developments in the area of FAIR data.

2.1.1 FAIR Guiding Principles

The FAIR guiding principles introduced by Wilkinson et al. [WDA+16] are intended to improve the infrastructure supporting the reuse of scholarly data with a focus on enhancing the ability of machines to automatically find and use the data. The publication describes the four foundational principles Findability, Accessibility, Interoperability and Reusability

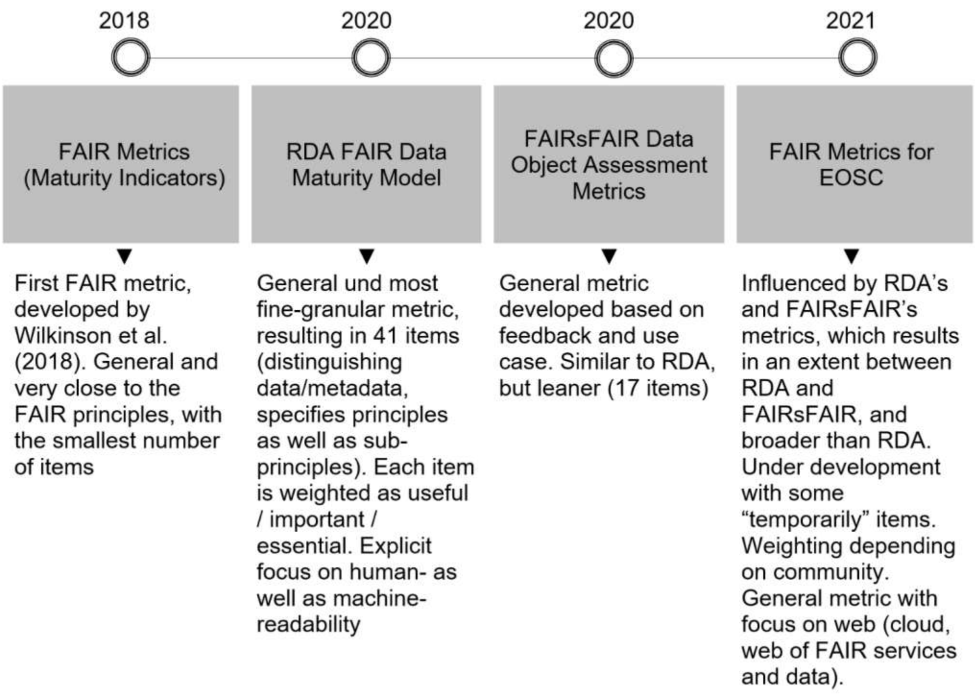

Figure 2.1: Simplified FAIR metrics overview [MWH+23].

and compared to other domain-specific publications relating to data management and archival, the FAIR principles aim to describe domain-independent high-level guidance to be applied on a wide variety of scholarly outputs. These high-level principles do not suggest specific implementation choices so that different stakeholders can adapt them to their need to fit a wide variety of technology choices and implementations.

Findability

The *Findability* principle concerns the location and possibility to access digital resources and consists of four sub-principles *F1, F2, F3, F4*. By providing accurate and comprehensive metadata and identifiers for digital objects, they can be discovered by both humans and machines. To achieve findability, data should be appropriately described with rich metadata, including information such as titles, descriptions, keywords, and identifiers. In addition, persistent and globally unique identifiers should be assigned to datasets to facilitate their citation and referencing.

F1 (meta)data are assigned a globally unique and persistent identifier

F2 data are described with rich metadata

F3 metadata clearly and explicitly include the identifier of the data it describes

F4 (meta)data are registered or indexed in a searchable resource

Accessibility

The *Accessibility* principle promotes the ease with which data can be accessed and obtained for both humans and machines and consists of two sub-principles *A1 and A2* where *A2* is further composed of *A1.1* and *A1.2*. By ensuring that data is available in a convenient and usable format, it can be retrieved and obtained by users with minimal barriers. Therefore, data should be made openly available whenever possible, or access should be granted through appropriate authorization mechanisms. Data should be provided in formats that are widely supported and easily readable by both humans and machines. In addition, access procedures, terms of use, and any necessary access controls should be clearly defined and communicated to users.

A1 (meta)data are retrievable by their identifier using a standardized communications protocol

- **A1.1** the protocol is open, free, and universally implementable
  - **A1.2** the protocol allows for an authentication and authorization procedure, where necessary

A2 metadata are accessible, even when the data are no longer available

Interoperability

The *Interoperability* principle promotes the ability of different systems, tools and resources to work together effectively and seamlessly, facilitating the exchange and integration of data, and consists of three sub-principles *I1, I2, I3*. By adopting and adhering to common standards, formats, and vocabularies for data representation, structure, and metadata, it is ensured that data can be combined and integrated from diverse sources, enabling meaningful analysis, comparison, and synthesis across different datasets and disciplines.

I1 (meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation

I2 (meta)data use vocabularies that follow FAIR principles

I3 (meta)data include qualified references to other (meta)data

Reusability

The *Reusability* principles refers to the extent to which data can be effectively and efficiently reused for different purposes beyond the original intention and consist of the sub-principle *R1* which is further divided into *R1.1, R1.2, R1.3*. It covers various aspects that make data usable and valuable for subsequent analysis, research, and application.

Priority	Findable	Accessible	Interoperable	Reusable	Total
Essential	7	8	0	5	20
Important	0	3	7	4	14
Useful	0	1	5	1	7
Total	7	12	12	10	41

Table 2.1: Distribution of FAIR indicators proposed by the RDA [Gro20].

R1 meta(data) are richly described with a plurality of accurate and relevant attributes

R1.1 (meta)data are released with a clear and accessible data usage license

R1.2 (meta)data are associated with detailed provenance

R1.3 (meta)data meet domain-relevant community standards

2.1.2 RDA Maturity Model

The FAIR principles defined by Mark Wilkinson [WDA+16] have to be considered as concepts instead of strict rules and may lead to diverse interpretations. In order to provide a unification of different interpretations of the FAIR principles, the RDA therefore established a Working Group (WG) titled *FAIR data maturity*. As a result, this WG presented a set of indicators and maturity levels for these indicators, as well as a set of guidelines and a checklist related to the implementation of these indicators [Gro20].

In total, the WG defined 41 indicators grouped into 3 different priority groups with the distribution of indicators depending on the corresponding FAIR principles and priority listed in Table 2.1. In addition to the description of the indicator, the WG also published corresponding assessment details to provide guidelines for the implementation of the indicators.

2.1.3 FAIRsFAIR Data Object Assessment Metrics

Devaraju et al. [DH21] developed object assessment metrics as part of the FAIRsFAIR project to evaluate the extent to which digital objects adhere to FAIR principles. In addition to the proposal of 17 metrics, listed in Table 2.2, they also published an automated assessment tool to apply these metrics in practice. The metrics are based on the indicators proposed by the RDA FAIR Data Maturity Model WG [Gro20] and the work of other project partners and have been adapted based on feedback provided by FAIR stakeholders. Compared to other FAIR guidelines that often rely on manual questionnaires and checklists, the FAIRsFAIR Data Object Assessment Metrics define practical tests and rationals for automated evaluation as well as a proof-of-concept evaluation of an automated FAIR evaluator titled F-UJI.

FAIR Principle	Metric ID	Description
F1	FsF-F1-01D	data are assigned a globally unique identifier
F1	FsF-F1-02D	data are assigned a persistent identifier
F2	FsF-F2-01M	metadata include descriptive core elements to support data findability
F3	FsF-F3-01M	metadata include the identifier of the data they describe
F4	FsF-F4-01M	metadata are offered in such a way that they can be retrieved by machines
A1	FsF-A1-01M	metadata contain access level and access conditions of the data
A1	FsF-A1-02M	metadata are accessible through a standardized communication protocol
A1	FsF-A1-03D	data are accessible through a standardized communication protocol
A2	FsF-A2-01M	metadata remain available, even if the data are no longer available
I1	FsF-I1-01M	metadata are represented using a formal knowledge representation language
I1	FsF-I1-02M	metadata use semantic resources
I3	FsF-I3-01M	metadata include links between the data and its related entities
R1	FsF-R1-01MD	metadata specify the content of the data
R1.1	FsF-R1.1-01M	metadata include license information under which data can be reused
R1.2	FsF-R1.2-01M	metadata include provenance information about data creation or generation
R1.3	FsF-R1.3-01M	metadata follow a standard recommended by the target research community of the data
R1.3	FsF-R1.3-02D	data are available in a file format recommended by the target research community

Table 2.2: FAIRsFAIR object assessment metrics [DH21].

2.1.4 FAIR Metrics for EOSC

The Recommendations on FAIR Metrics for EOSC [fRaIECBA+21] investigates developments in the field of FAIR data and evaluates them in the context of the European Open Science Cloud. In particular, the FAIRsFAIR project and the RDA maturity model have been analyzed. As a result, the report proposes a set of recommendations on how FAIR metrics should be implemented at the European and international levels.

2.1.5 FAIR Evaluators

To use FAIR metrics in practice tools and resources have been proposed to produce FAIR measurements corresponding to the given FAIR metrics. FAIRassist [1] publishes a list of currently availableFAIR evaluation resources, pictured in Table 2.3, distinguished between *automated*, *semi-automated*, *questionnaires* and *checklists*. Furthermore, Krans et al. [KAN+22] investigate the use of FAIR evaluation tools based on the experience of individual researchers and give recommendations for both users and developers of FAIR assessment tools.

An issue in the field of FAIR metrics and evaluators, as identified by Wilkinson et al. [WSM+22] is the inconsistency of the results between different FAIR evaluators. Not only from a technical standpoint does the format of results of different automated solutions differ but different tools also provide different opinions on the same digital objects which leads to frustration and confusion. This is due to the difference in the underlying definition of FAIR metrics as well as to differences in the way the evaluators are implemented. Moser et al. [MWH+23] provided a mapping, listed in Figure 2.2, from different metrics to the evaluators that use them. It provides an overview of how popular evaluators use different FAIR metrics that are nevertheless all derived from the FAIR guiding principles.

2.1.6 FAIR Enabling Services

FAIR enabling services and repositories are essential for the development of FAIR evaluators as they provide necessary information for the automated evaluation. According to Devaraju et al. [DMC+21] such services could provide information of qualified repositories, metadata standards, licenses, and policy registries. FAIRsharing [SMRS+19] is such a FAIR enabling service and provides interlinked information on standards, databases, and policies. By interlinking these concepts, FAIR evaluators can automate certain tests by comparing metadata of a digital resource with the curated information available on FAIRsharing.

[1] https://fairassist.org/

Resource	Automated	Semi Automated	Questionnaire	Checklist
5 Star Data Rating Tool			✓	
AutoFAIR		✓		
CLARIN Metadata Curation Dashboard	✓			
Data Stewardship Wizard			✓	
Do I-PASS for FAIR				✓
F-UJI	✓			
FAIR Data Self-Assessment Tool			✓	
FAIR Evaluator	✓			
FAIR Evaluation Services	✓			
FAIR data Self-Assessment Tool				✓
FAIR enough	✓			
FAIR-Aware			✓	
FAIR-Checker	✓			
FAIRdat			✓	
FAIRness self-assessment grids				✓
FAIRshake	✓		✓	
FOOPS!	✓			
FairDataBR			✓	
GARDIAN FAIR Metrics				✓
HowFAIRIs	✓			
O'FAIRe	✓			
OpenAIRE Validator - FAIR assessment	✓			
PresQT (Preservation Quality Tool)	✓			
RDA Maturity Model				✓
TRIPLE Training Toolkit			✓	✓

Table 2.3: Resources to measure and improve FAIRness [fai].

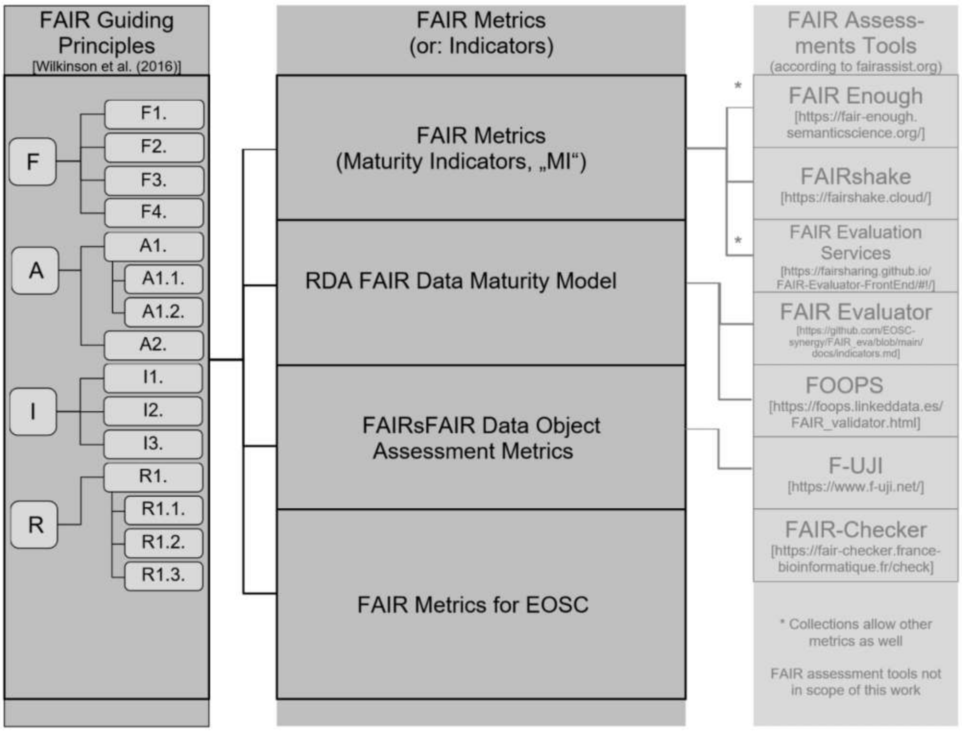

Figure 2.2: FAIR metrics identified by usage in FAIR assessment tools [MWH+23].

2.2 Data Quality Dimensions

While FAIR metrics are concerned with the *findability*, *accessibility*, *interoperability*, and *reproducibility* of digital objects, depending on the domain and technology in question, there exists a vast variety of data quality standards. For the field of DMPs, to our knowledge, there currently exists no literature on quality metrics, but for other fields such as LD and data catalogs, previous work has been done that can be adapted to be applied on DMPs. To that extent, Zaveri et al. [ZRM+15] conducted a systematic review of approaches to assess the quality of LD. Following this summary, Martinez-Gill [MG23] gives an overview of different data quality metrics for use in data catalogs, including the one proposed by Zaveri et al. [ZRM+15] and ISO/IEC 25012 [iso] and proposed an alignment. Extending this alignment we added the DMP review goals proposed by Miksa et al. [MSS+23] and FAIR principles as pictured in Table 2.4. In the remainder of this section, the corresponding alignments of the data quality dimensions concerning the DMP review goals mentioned by Miksa et al. [MSS+23] will be put into context with other proposed quality dimensions and further explained.

Zaveri et al. [ZRM+15]	**ISO 25012** [iso]	**Martinez-Gil** [MG23]	**Miksa et al.** [MSS+23]	**Wilkinson et al.** [WDA+16]
Accuracy	Accuracy	Accuracy	Feasibility	-
Completeness	Completeness	Completeness	Completeness	-
Consistency	Consistency	Consistency	Feasibility	-
Timeliness	Currentness	Timeliness	-	-
Availability	Availability	-	Feasibility	-
Accessibility	Accessibility	-	Quality of Actions	Accessibility
-	Portability	Licensing	Quality of Actions	Accessibility, Reusability, Interoperability
Understandability	Understandability	Readability	Quality of Actions	Interoperability
-	Compliance	-	Guideline Compliance	-
-	-	Compatibility	Quality of Actions	Reusability, Interoperability
-	-	Similarity	-	-

Table 2.4: Alignment of quality metrics proposed in the literature adapted from Martinez-Gil [MG23].

2.2.1 Completeness

Miksa et al. describe the goal *Completeness* as "the coverage of all relevant aspects of research data management in the DMP" [MSS+23]. In ISO/IEC 25012 [iso] *Completeness* is defined as "the degree to which subject data associated with an entity has values for all expected attributes and related entity instances in a specific context of use" [iso]. The corresponding definition by Zaveri et al. is "The degree to which all required information is present in a particular dataset" [ZRM+15].

2.2.2 Feasibility

Miksa et al. describe *Feasibility* as "the possibility to put all DMP content into practice through concrete actions" [MSS+23]. There is no one-to-one mapping to the dimensions listed in Table 2.4, but the quality metrics *Availability*, *Consistency* and *Accuracy* cover parts of the feasibility dimension.

Availability

In ISO/IEC 25012 [iso] *Availability* is defined as "The degree to which data has attributes that enable it to be retrieved by authorized users and/or applications in a specific context of use." [iso]. Similarly, Zaveri et al. define it as "the extent to which data is present, obtainable, and ready for use."

Consistency

In ISO/IEC 25012 [iso] *Consistency* is defined as "The degree to which data has attributes that are free from contradiction and are coherent with other data in a specific context of use." [iso]. Zaveri et al. provide a definition more tailored to LD: "Consistency means that a knowledge base is free of (logical/formal) contradictions concerning particular knowledge representation and inference mechanisms" [ZRM+15]. Compared to the related dimension of *Compliance*, *Consistency* is only concerned with finding logical errors, while *Compliance* refers to the verification of constraints regarding the semantics of the data.

Accuracy

In ISO/IEC 25012 [iso] *Accuracy* is defined as "the degree to which data has attributes that correctly represent the true value of the intended attribute of a concept or event in a specific context of use." [iso]. Zaveri et al. coin the narrower term of *Semantic Accuracy* as "the degree to which data values correctly represent the real world facts" [ZRM+15].

2.2.3 Quality of Actions

Miksa et al. define *Quality of Actions* as "the relevance and effectiveness of the actions listed in the DMP and performed according to it" [MSS+23] and "assess whether what was described can be or was (depending on the phase) implemented according to community standards" [MSS+23]. Aligning dimensions from Table 2.4 are *Accessibility*, *Portability Licensing* and *Compatibility*. These dimensions can be aligned with the FAIR principles introduced in Section 2.1.

2.2.4 Non-ambiguity

Miksa et al. [MSS+23] define this goal with a DMP having clear and non-ambiguous formulation, especially concerning the non-machine-actionable parts. Table 2.4 shows an alignment with the *Understandability* dimension from Zaveri et al. [ZRM+15] and ISO 25012 [iso] and the *Readability* dimension from Martinez-Gil [MG23].

2.2.5 Guideline Compliance

Miksa et al. [MSS+23] mention the fulfillment of funder requirements as a review goal. In ISO/IEC 25012 [iso] *Compliance* is defined as "The degree to which data has attributes that adhere to standards, conventions or regulations in force and similar rules relating to data quality in a specific context of use" [iso]. Foidl et al. [FBM21] propose to use the conventions published by Science Europe [Eur21] as a guideline for maDMPs.

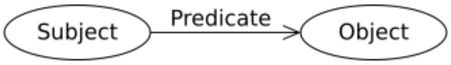

Figure 2.3: RDF overview [CHWL14].

2.3 Semantic Web

The Semantic Web [BLHL23] is a concept for organizing and linking data on the Web in a way that allows computers to understand and process it more efficiently. Standards such as RDF and OWL are widely used, and the DCSO standard proposed by the RDA [CCE+22] is expressed in the form of RDF triples as well. Previous work on maDMP evaluation used this ontology and queried the graph representing a maDMP using SPARQL [FBM21]. Recently, however, new technologies have emerged that are tailored towards the use for validation of the semantics of these data structures [PK22, RLH22] and with SHACL[2] there also exists a W3C-approved standard for the validation of RDF data.

2.3.1 Resource Description Framework

The Resource Description Framework (RDF) is a technology used to represent information on the Web. It provides a standard way to describe resources and their relationships, making it possible to share and integrate data across different applications, and is a foundational technology in the W3C's Semantic Web stack [CHWL14].

The basic building block of RDF is the triple, as shown in Figure 2.3. It consists of three parts: subject, predicate, and object. The subject is the resource being described, the predicate is the property or attribute of the resource, and the object is a value or another resource. A resource in RDF is identified by a URI and can represent an arbitrary real-world or abstract entity. In addition to resources, RDF can represent literal values in the form of various data types.

2.3.2 OWL

Web Ontology Language (OWL) is a language for authoring ontologies and is built on top of RDF. OWL is is part of the W3C's Semantic Web technology stack and provides a way to formally represent the meaning of terms and the relationships between them, allowing machines to understand and reason about the semantics of information.

According to McGuiness et al.[MVH04], key characteristics include *ontologies*, which are formal specifications of terms and relationships within a particular domain, *classes* which represent sets of individuals and *instances* being specific members of those classes. Furthermore, OWL supports the definition of properties, which represent relationships between individuals. Properties can be used to model attributes, associations, and other connections between entities. As a result, it enables automated reasoning, so that OWL

[2] `https://www.w3.org/TR/shacl/`

ontologies can be used to infer new information, which enables machines to make logical deductions on a given dataset. OWL operates under the open-world assumption, meaning that the absence of information does not imply falsehood.

2.3.3 SPARQL

SPARQL Protocol and RDF Query Language (SPARQL) is a query language designed for querying and manipulating RDF data with a syntax similar to SQL but adapted for RDF graphs. SPARQL queries are constructed using triple patterns, which consist of subject-predicate-object combinations to match specific parts of RDF triples in the dataset. SPARQL also supports aggregate functions to perform calculations on the results of the query.

2.3.4 SHACL

SHACL is a W3C recommendation for the validation of RDF graphs. SHACL shapes are templates specifying the structure of RDF graphs and define the presence and characteristics of nodes and arcs. It includes node constraints, property constraints, value constraints, and complex constraints that ensure that a RDF graph meets the intended structural and semantic requirements. SHACL can also be used for inferencing, resulting in the deduction of additional information based on the specified shapes and constraints.

2.3.5 Data Quality Indicator Frameworks

Besides FAIR evaluators there also exist tools for the evaluation of data structures for other domains. Especially the field of evaluation of the quality of linked data is significant for this thesis, as the availability of the DCSO DMPs can be interpreted and evaluated with respect to linked data quality metrics. Zhang et al. [ZBC23] give an overview of the existing approaches of RDF validation, listed in Table 2.5 which includes their own proposal[3], Luzzu[4] and RDFUnit[5]. Besides those frameworks in the comparison, there exist a wide variety of different metrics and frameworks such as the Piveau Metrics [WKJ+23, KSD+20] for evaluation of the quality of open data catalogs and SemQuire [LSGG18] for the assessment of linked data, DQFIRD [XXWT16] for quality-based ranking of datasets, and a framework by Weikopf et al. [WW13] for the assessment of health record data quality.

The approaches are manifold and sometimes tailored to a specific domain, but as the comparison by Zahng et al. [ZBC23] in Table 2.5 shows that semantic web technologies are widely used in the evaluation of data quality.

[3] https://github.com/sxzhang1201/assess-rdf-resource
[4] https://github.com/Luzzu/Framework
[5] https://github.com/AKSW/RDFUnit

	RDFUnit	Luzzu	Zhang et al [ZBC23]
Purpose	Proposal of a framework based on the data quality integrity constraints (represented in SPARQL patterns)	Proposal of a semantic framework based on Dataset Quality Ontology (daQ)	Proposal of an automated approach to assess the foundational characteristics as the starting point for linked data quality assessment
Novelty	Use of SPARQL pattern	Add semantic layer and quality metadata to assessment framework	Focus on foundational aspects

Table 2.5: Comparison of Data Quality Evaluation Frameworks by Zhang et al. [ZBC23].

2.3.6 Data Quality Vocabulary

The Data Quality Vocabulary (DQV) developed by the Data Web Best Practice working group of W3C [AI20] is a RDF vocabulary for describing data quality information. Figure 2.4 shows the main components and their relationships with each other. Most of the proposed concepts are adapted from existing vocabularies such as *DCTERMS* [6] for interoperable metadata and *PROV*[7] to include provenance information.

DQV itself does not propose a specific definition of quality metrics, but rather a framework to define metrics and their dependencies depending on the use case. In this regard, it defines the four main entities *QualityMeasurement*, *Metrics*, *Dimension* and *Category* as listed in Table 2.6.

2.4 Scientific Knowledge Graphs

Knowledge graphs are networks of entities and relationships, usually expressed in W3C standards such as OWL and RDF. The term Scientific Knowledge Graph (SKG) in particular refers to infrastructures representing scholarly knowledge in a structured, interlinked and semantically rich way describing various actors, documents, and research knowledge, as well as their reciprocal relationships [skg]. Additionally, services, such as the OpenAire Research Graph [ope] and the Dataset Knowledge Graph [FL21] aggregate information from different resources into a uniform SKG to provide a standardized gateway to research information.

In addition to the OpenAire SKG and Dataset Knowledge Graph, there exists a wide variety of SKGs covering different categories and needs. But, as Aryani et al.[AFM+20] point out, this fragmentation also poses challenges when working with them. The key challenge they point out is the lack of a classification framework and an interoperability

[6]https://www.dublincore.org/specifications/dublin-core/dcmi-terms/
[7]https://www.w3.org/TR/prov-o/

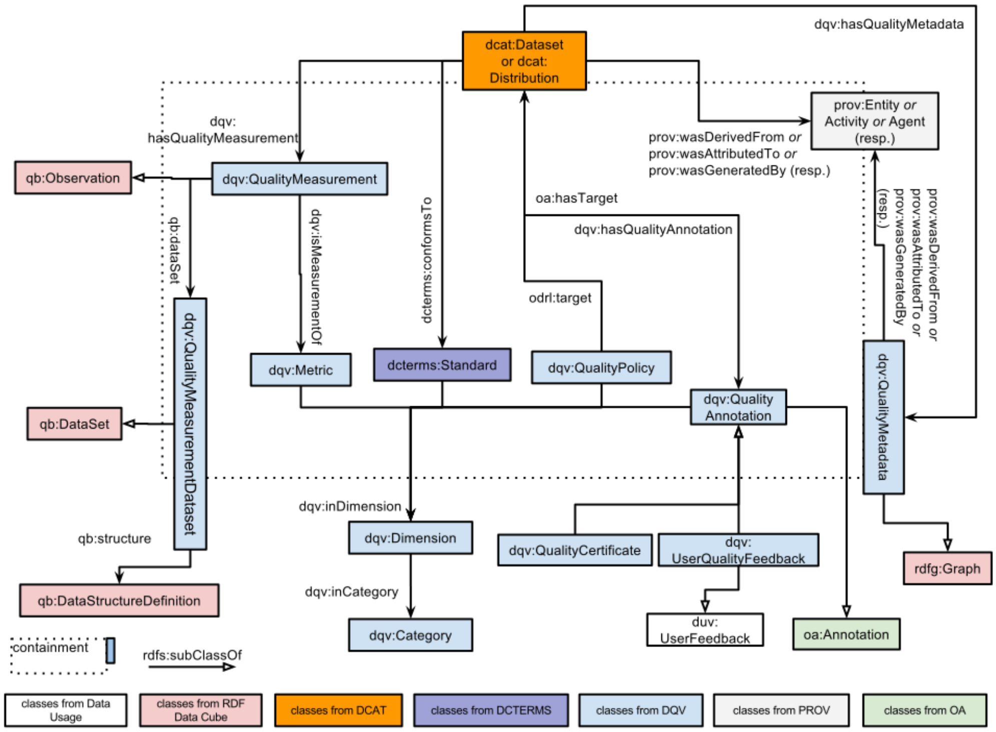

Figure 2.4: Data Quality Vocabulary (DQV) [AI20].

RDF Class	Definition
dqv:QualityMeasurement	The result of evaluating a dataset against a specific metric. It includes the value of the metric and any associated information.
dqv:Metric	A quantitative measure or characteristic used to express a specific aspect of data quality.
dqv:Dimension	An aspect or category of data quality that can be measured using one or more metrics. Dimensions provide a way to organize and categorize metrics.
dqv:Category	Represents a group of quality dimensions in which a common type of information is used as quality indicator.

Table 2.6: DQV classes and definitions [AI20].

framework to foster seamless exchange between various initiatives. Figure 2.5 shows as a result of this initiative the classification of different knowledge graphs based on available research entities, their applications, connected data sources, added value, data export possibilities, FAIRness and openness. The SKGs included in the comparison include, in addition to the OpenAire Research Graph, the PID graph [8], the Open Research Knowledge Graph (ORKG) [9] the Research Graph [10] and ScholeXplorer [11].

2.5 Data Management Plans

DMPs are becoming increasingly important for various stakeholders involved in RDM [Mic15, WBNZ17, HVMS19]. The general goal of a DMP is to document and describe how data will be handled during a project and how it will be treated after the project ends. As such, they typically cover various stages in the data life-cycle beginning with data discovery and the successive stages of collection and organization, quality assurance, documentation until data preservation and sharing with others. Michener et al. [Mic15] align the research with the data life-cycle, as shown in Figure 2.6. Depending on the corresponding step in the research cycle, in the data life-cycle researchers typically outline how data will be managed during and after the project, discover and acquire existing data, collect and organize new data, assure the quality of the data, describe the data with metadata, use the data in their research, and lastly preserve and share the data with others. For each step in the data lifecycle (B), a yellow circle is added providing references to the 10 rules for creating a good DMP [Mic15].

Depending on the given requirements of stakeholders, different data life-cycle models exist, as outlined by Ball et al. [Bal12] in their review of data management life-cycle models. Popular models include the DCC Curation Lifecycle Model, the I2S2 Idealized Scientific Research Activity Lifecycle Model, and the DDI Combined Life Cycle Model. What they have in common is that they provide a structure for the many operations that are being performed during the different stages a data record undergoes in its lifetime. This structure makes it easier to prepare many curatorial actions because they have been explicitly prepared for in advance. Furthermore, certain funders and other funding bodies require a DMP to be actively updated during the different stages of the data life-cycle and require a copy of this document at different stages of the research cycle, such as for grant approval.

2.5.1 Machine Actionable Data Management Plans

As outlined in Section 1.1, DMPs are often a requirement by funding bodies when applying for a research grant. But the utility of traditional DMPs is limited because they are usually static documents answering questions from a given template and therefore

[8] `https://www.project-freya.eu/en/pid-graph/the-pid-graph`
[9] `https://orkg.org/`
[10] `https://researchgraph.org/`
[11] `https://scholexplorer.openaire.eu`

	Research Entities	Applications	Data Sources	Added value	Data Export	FAIRness	Openness
PID graph	**PIDs:** DOI, ORCID, ROR, Crossref Funder ID **Entities**: publications, datasets, software, people, research organizations, funders, research data repository registries	Discovery, Research Impact, Open Science Monitor, Brokering, Reporting to funders, Statistics	PID providers	Standard GraphQL query interface with client libraries available in many languages. Strong support for Jupyter notebooks	**API**: GraphQL	**Findable:** Graphs are searchable by PIDs and metadata with GraphQL API **Accessible**: access described entities via PID **Interoperable**: Input: DataCite XML, Crossref XML, ORCID XML. Output: GraphQL JSON **Reusable**: Graphs are reusable	Redistributed free of charge
OpenAIRE Research Graph	**PIDs:** DOI, ORCID, accession numbers, PMCID, URLs, MAG IDs **Entities:** Literature, datasets, software, funders, grants, organizations, researchers, data sources **Relationships:** DataCite relationships, fundedBy, similarTo, hasAuthoraffiliatedWith	Discovery, Research Impact assessment, Open Science Monitoring, Brokering, Reporting to funders, Statistics	Any data source trusted by scientists: repositories, archives, registries, databases, publishers	Enrichment of metadata and relationships by full-text mining, User-feedback, Inference by context propagation, deduplication, Provenance tracking	**Format:** OpenAIRE XML format **APIs:** LOD, OAI-PMH, Dumps, REST Search APIs	**Findable:** searchable on Zenodo, accessible by DOI **Accessible**: every entity is openly accessible via HTTP API and dump **Interoperable**: OpenAIRE XML, documented online **Reusable**: Content can be reused with CC-BY licence	Redistributed free of charge under CC-BY licence
ORKG	**PIDs:** DOI, ORCID, URLs **Entities:** Literature and items of its content **Relationships:** addressed problems, utilized materials, employed methods, yielded results.	At the granularity of items of scholarly literature content, discovery, comparison, recommendation, visualization, reuse	Literature, research data repositories, terminologies, LIMS/ELN	Multimodal infrastructure for the acquisition, curation and publishing of machine-actionable scholarly knowledge.	**Format:** JSON and RDF serializations **APIs:** REST API, SPARQL	**Findable:** ORKG search, but currently lacking findability through 3rd party systems. Plans to assign DataCite DOIs to elements of ORKG content. All resources are URL-identified. Neo4j dumps or RDF exports can be deposited in a suitable data repository. **Accessible:** All ORKG resources are URL-retrievable, accessible and have their own descriptive landing page. Content can be accessed programmatically via REST API. **Interoperable:** Data use a graph-based data model, but currently lacking formal semantics. Alignment with external terminologies is technically possible but not broadly practised. **Reusable:** Content is reusable under CC BY-SA licence, provenance is tracked. Neo4j dumps or RDF exports can be made available for reuse.	Data and Software are released under CC BY-SA and MIT licences, respectively.
Research Graph	**PIDs:** DOI, ORCID, PURL, ISNI, GRID, PMCID, Scopus_ID **Entities:** academic articles and grey literature, datasets, funders, grants, organizations, researchers, datasets **Relationships:** authorship, funding, citation, usage, known_as, employment, custodian, management, etc.	Supporting repositories and research infrastructures	PID providers, data repositories, publishers, funders, discovery services and aggregators	Identity resolution, metadata enhancement, topic modelling, clustering, text mining and GIS mapping	**Format**: XML and JSON **APIs:** Cloud Hosted Services, REST API and GraphQL	**Findable:** Metadata available via researchgraph.org **Accessible:** Controlled Access available **Interoperable:** Input: RDF, DDI, RIF-CS, Dublin Core, Scholix, DataCite, Crossref, and many other metadata formats. Output: RDF and Research Graph Schema (JSON, XML) **Reusable:** Includes subgraph reusable under CC-BY licence while some other parts only accessible for limited use under NC-ND-SA-Creative Common	Controlled Access
Scholexplorer	**PIDs:** DOI, accession numbers, PMCID, URLs **Entities:** Literature, datasets **Relationships:** citedBy, supplementedBy, references	Discovery, Statistics, PID Resolution	DataCite, CrossRef Event Data, EMBL-EBI (EPMC), OpenAIRE, Scholix-compliant data sources	Deduplication by PID, provenance tracking	**Format:** Scholix [?] **APIs:** Dumps on Zenodo, REST Search APIs	**Findable:** searchable on Zenodo, accessible by DOI **Accessible:** every entity is openly accessible via REST API and dump **Interoperable:** Scholix links **Reusable:** Content is released under CC-BY licence	Redistributed free of charge under CC-BY licence

Figure 2.5: Classification of SKG initiatives [AFM+20].

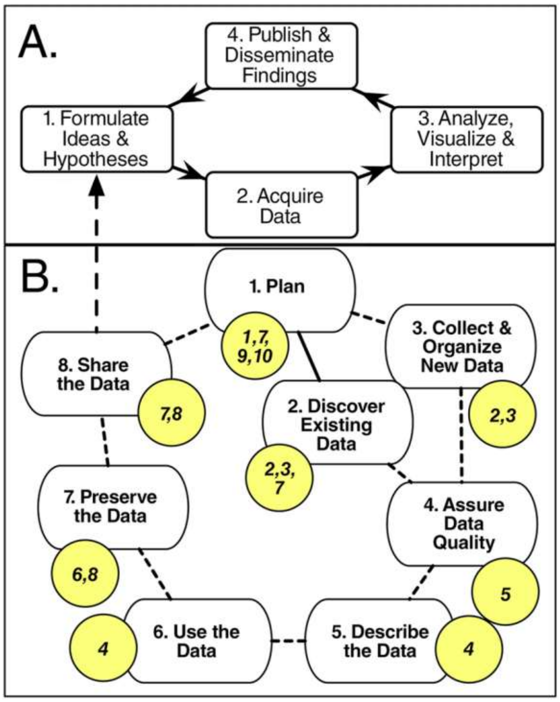

Figure 2.6: Relationship of the research life cycle (A) to the data life cycle (B) [Mic15].

are not machine-actionable. As a result, they do not provide the benefits discussed in Section 2.5 regarding integration and automation of research data management and have often been seen as a burden and additional paperwork by researchers. The Research Data Alliance (RDA) recognized the importance of providing a standard to make DMPs machine actionable so that systems can exchange and act on information about the data used and produced by researchers, and proposed ten principles, shown in Figure 2.7, to put maDMPs into practice and realize their benefits. [MWN+21].

Since the proposal of machine-actionable DMPs [MSMJ19, SJMM17] and the publication of the corresponding application profile by Miksa et al. [MWN+21] work has been conducted to utilize machine actionability. As such, different tools, summarized and compared by Jones et al. [JPH+20], have been published that help researchers create and manage DMPs while other solution aim to use DMPs as a central information hub in the management of research data [MOR22, OMK, MRGB17].

RDA DMP Common Standard Application Profile DCS

Miksa et al. [MWN+21] describe the DCS application profile for maDMPs which has been developed by the RDA DMP Common Standards Working Group, with the goal of representing information over the entire DMP life-cycle in a machine actionable format. Figure 2.8 shows the main concepts used in the application profile. Each of these concepts contains additional properties, as can be seen in Listing 2.1, which presents an

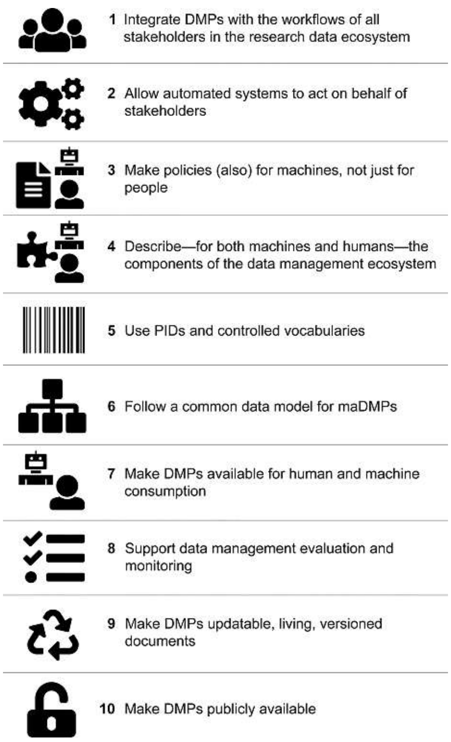

Figure 2.7: Ten principles for machine-actionable data management plans [MSMJ19].

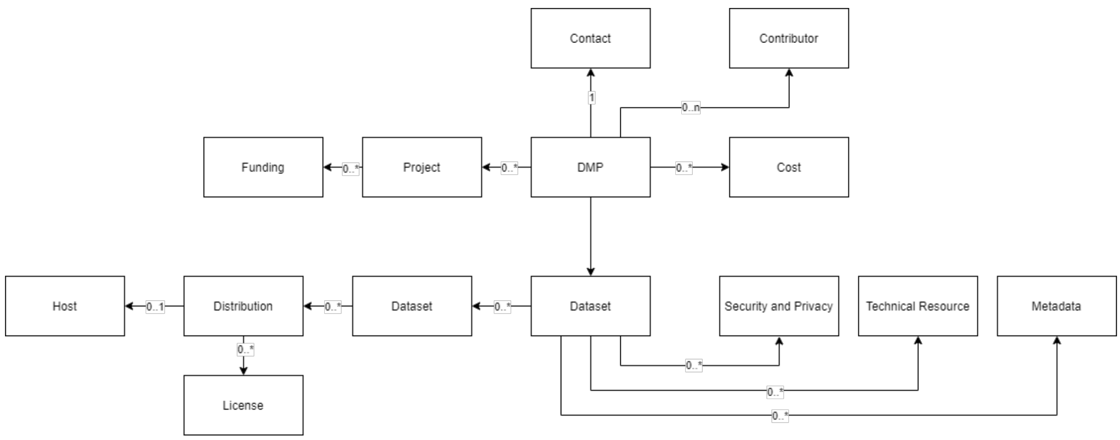

Figure 2.8: Application Profile for maDMPs [MWN+21].

example minimal maDMP in JSON notation that is compliant with the DCS application profile. Each concept defined in the application profile can contain other concepts in the specified multiplicities with the *DMP* concept being the root entity. Furthermore, the application profile defines the necessary properties in a concept, together with the data type, multiplicities, and allowed values. Where possible, those values follow some standardized controlled vocabulary.

```json
{
  "dmp": {
    "title": "Minimal DMP",
    "contact": {
      "contact_id": {
        "identifier": "0000-0002-5164-2690",
        "type": "orcid"
      },
      "mbox": "moritz.staudinger@tuwien.ac.at",
      "name": "Moritz Staudinger"
    },
    "created": "2023-10-23T10:10:23.6",
    "dmp_id": {
      "identifier": "10.5281/zenodo.13119047",
      "type": "doi"
    },
    "dataset": [
      {
        "dataset_id": {
          "identifier": "10.5281/zenodo.4699026",
          "type": "doi"
        },
        "title": "Results: Analysis of Performances of kNN
            and Random Forest",
        "personal_data": "unknown",
        "sensitive_data": "unknown"
      }
    ],
    "ethical_issues_exist": "unknown",
    "language": "eng",
    "modified": "2023-10-23T10:10:23.6"
  }
}
```

Listing 2.1: Example of a minimal DCS compliant maDMP in JSON notation.

The complete application profile including the full definition of concepts and detailed specification on included properties, allowed values from controlled vocabulary, as well as their data types and multiplicity is published in the repository of the work group [12].

Persistent identifiers (PIDs)

Miksa et al. [MSMJ19] propose 10 principles to put maDMPs into practice and realise their benefits. As stated in principle 5, maDMPs should make use of PIDs and controlled

[12] https://github.com/RDA-DMP-Common/RDA-DMP-Common-Standard

vocabularies wherever possible. PIDs provide a way to identify and locate resources and can refer to people and publications, as well as datasets, file types, repositories, organizations, policies and other elements of the research data ecosystem. The benefit of those persistent identifiers is that they are a permanent, unique reference to a digital resource. Compared to other resource locators, persistent identifiers promise to reliably provide references to a digital entity that do not break over time. Registration agencies, such as Crossref [13] and DataCite [14] for DOIs, register the identifiers together with metadata and citation information [pid].

DSCO Ontology of the RDA DMP Common Standard

The DMP Common Standard (DCS) by the RDA is a technology agnostic specification and does provide any recommendation on underlying technologies. As a result, there does not exist an explicit specification of the underlying data model, as well as instances of recommended controlled vocabularies. Lastly, there is no clear recommendation on how to provide extension while distinguishing between core specification and extension.

To address these points, Cardoso et al. [CCE+22] propose the DMP Common Standard Ontology (DCSO), a RDF serialization of the DCS application profile. With this implementation, semantic web technologies can be used for validation, and SHEX validation is provided for the ontology [CPB20].

DCS Funder Extension

The DCS application profile has originally been developed as an information exchange format. Because of that initial requirement, the standard is intentionally kept as minimal as possible and only contains information deemed essential for its purpose. But in the initial proposal also the need for a broader range of use cases has been recognized and the possibility of extensions has been discussed. With the introduction of DCSO there exists an implementation of the application profile that naturally enables the integration of extensions through the use of namespaces to adapt a DCS maDMP to various use-cases.

Cardoso et al. [CJM+20] proposed an extension with the aim of increasing coverage on the most prominent DMP funder templates as a result of the RDA Hackathon on machine-actionable Data Management Plans that took place between 27th and 29th May 2020. They considered the H2020 DMP Template[15], the Science Europe Core Requirements for Data Management Plans[16], the National Science Foundation (NSF) GENERIC Core Requirements for DMPs Questions[17] and the U.S. Geological Survey

[13] https://www.crossref.org/
[14] https://datacite.org/
[15] https://ec.europa.eu/research/participants/data/ref/h2020/other/gm/reporting/h2020-tpl-oa-data-mgt-plan-annotated_en.pdf
[16] https://www.scienceeurope.org/media/jezkhnoo/se_rdm_practical_guide_final.pdf
[17] https://www.nsf.gov/pubs/policydocs/pappg19_1/pappg_2.jsp#IIC2j

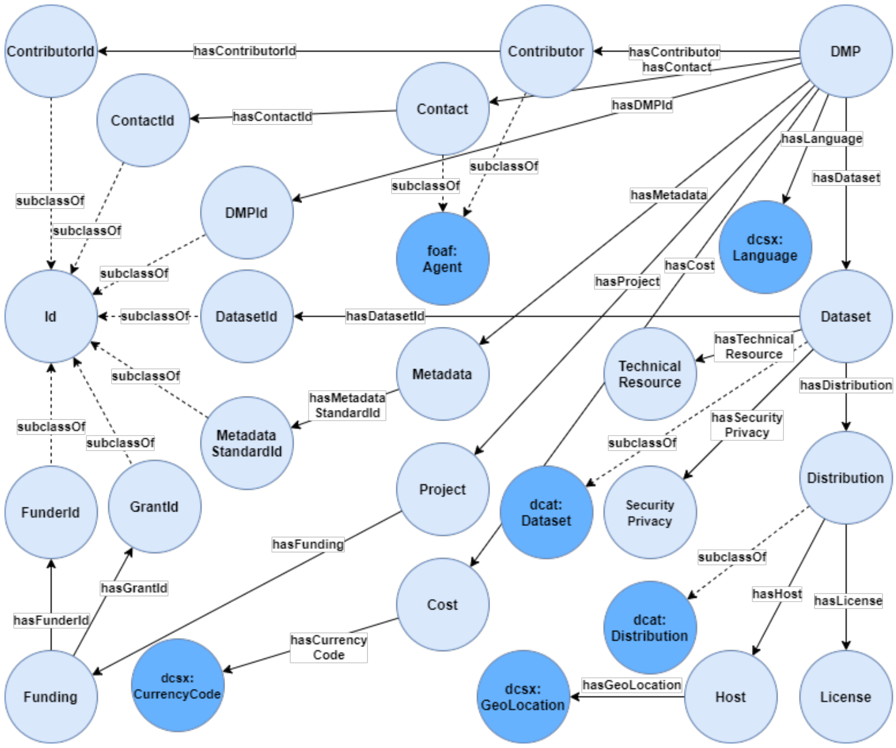

Figure 2.9: DCSO Ontology [CCE+22].

(USGS) Data Management Plan Review Checklist[18]. The recommendations resulting from this hackathon are summarized in this section.

Proposed Changes of the DCS Standard Table 2.7 shows the proposed changes of properties in the entity *dataset*. The field *data_quality_assurance* has been changed from a String to a nested data structure *data_quality_assurance*, listed in table 2.8 with carnality 0..n. This new entity provides a field to describe the method used in the quality process, as well as 0..n references to external data quality assurance artifacts indicated by *data_quality_assurance_id*.

Proposed Additions to the DCS Standard To further cover the requirements of the investigated funders' DMP templates additional objects, listed in Table 2.9, have been proposed to be added to the suggested funder extension. The additions include a data structure *related_policy* to refer policies relevant for the DMP, additional properties appended to the entry *dataset* to specify if the given dataset is reused as well as its target audience and a reference to a resource providing information about the employed methodology in the scope of the dataset. In addition, extensions have been proposed for the entry *distribution* to provide an explanation if the data cannot be shared openly.

[18] https://www.usgs.gov/media/files/usgs-data-management-plan-review-checklist

Name	Description	Previous Data Type	New Data Type	Multiplicity
Dataset				
data_quality _assurance	Provide any information on the measures taken during the research process to ensure the data quality.	String	Nested Data Structure	0..n

Table 2.7: Proposed changes of properties in *dataset* [CJM+20].

Name	Description	Data Type	Multiplicity
data_quality_assurance			
description	Free text to describe a method used in the data quality process.	String	1
data_quality _assurance_id	Identifier for a Data Quality Assurance artifact	Nested Data-Structure	0..1
data_quality_assurance_id			
identifier	A unique identifier for a Data Quality Assurance artifact	String	1
type	Identifier type	String	1

Table 2.8: Proposed additions to accommodate changes of the field *data_quality _assurance* [CJM+20].

2.6 Discussion

In this chapter, we discussed related work in the scope of automated indicators for the evaluation of DMPs as background for this thesis. As DMPs relate to a wide range of RDM activities, a solution for the automated evaluation of DMPs requires knowledge of a wide range of different topics. Among the included topics, we discussed methods and standards available to measure the quality aspects of digital objects with a focus on research data. To this extent, we outlined recent developments in the field of FAIR data as well as more generic data quality initiatives such as DQV which is used mainly by linked data evaluators to measure quality metrics with respect to data quality dimensions proposed in ISO/IEC 25012. Despite a common underlying use case, publications related to DQV [AI20] and FAIR data [WDA+16] use a different terminology. While DQV introduces the terms *Category*, *Dimension*, *Metric* and *Measurement*, the FAIR community uses the terms *FAIR Principle*, *Indicator* and *Evaluation Method*. In this discussion, we therefore provide an alignment between those terms in order to provide a common ground to enable the integration of both fields of quality evaluation into one coherent framework.

Name	Description	Data Type	Multiplicity
dmp			
related_policy	To link to all documents needed to be compliant to requirements within this DMP (e.g. legal, ethical, contractual, guidelines, procedures, standards,...)	Nested Data Structure	0..n
related_policy			
description	Description	String	1
related_policy_id	Related policy ID	Nested Data Structure	1
related_policy_id			
identifier	Identifier of the document	String	1
type	Type (e.g. URL, DOI)	Nested Data Structure	1
dataset			
is_reused	To explicitly indicate whether the dataset is reused or was produced in the course of research. Allowed values are: reused, produced.	Term from Controlled Vocabulary	1
target_audience	To state for whom this dataset can be relevant.	String	0..1
methodology	To describe methodology, procedures, workflows, etc. on how the dataset is created, can be recovered, ...	Nested Data Structure	0..1
methodology			
description	Methodology, procedures, workflows, etc.	String	1
methodology_id	Identifier for a methodology artifact	Nested Data Structure	1
methodology_id			
identifier	A unique identifier for a methodology artifact	String	1
type	Identifier type	String	1
distribution			
restriction _explanation	To describe any reason why data cannot be shared openly.	String	0..n

Table 2.9: Proposed additions to DCS to increase coverage of selected funder's DMP guidelines [CJM+20].

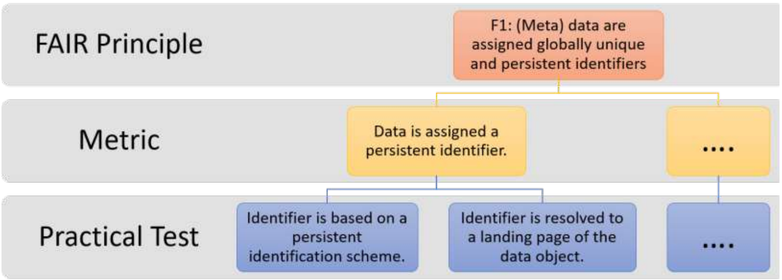

Figure 2.10: Example of principles, metrics and tests as used in the Horizon 2020 FAIRsFAIR project [DH21].

Figure 2.10 shows the hierarchy of terms used by the Horizon 2020 FAIRsFAIR project, which implements the FAIR principles outlined in Section 2.1. In this hierarchy, a *FAIR Principle* groups *Metrics* and these metrics are further described by *Practical Tests*. In more recent publications on FAIR principles the term *Metric* has been replaced with *Maturity Indicator* MI. Wilkinson et al. [WDS+19] explain their purpose as to "describe facets of FAIRness that can be objectively evaluated by a machine". Furthermore, the RDA states, that *Indicators* aim to measure the state or level of a digital resource concerning a specific FAIR *Principle* [Gro20].

The terms coined by DQV [AI20] and the RDA [Gro20], although different, describe the same aspects in the evaluation and description of aspects of quality of digital objects. The W3C DQV, presented in Section 2.3.6, uses the term *Dimension* to describe a level of abstraction relatable to FAIR *Principles* and the term *Metrics* that can be compared to FAIR *Metrics* or *Maturity Indicators*. In this thesis, we use the definitions of *Metric* and *Measurement* from Albertoni et al. [AI20]. A *Metric* represents a standard to measure a *Quality Dimension*, where a *Quality Dimension* represents criteria relevant for assessing a quality aspect. A *Measurement* represents the result of a given dataset against a specific metric.

In addition to these aligned terms, there are also concepts in DQV that are not covered by FAIR and vice versa. DQV is more focused on providing a structure to store measured metrics, while the FAIR community is more focused on describing what should be tested and less focus is placed on the subsequent result. As a consequence the term *Category* from DQV has no matching counterpart in the vocabulary used by FAIR evaluators and the FAIRsFAIR project uses *Practical Tests* to further define how metrics and their corresponding measurements are composed. Furthermore, the concept of defining tests is not present in DQV but appears in some implementations of FAIR evaluators, such as F-UJI.

Table 2.10 summarizes the alignment between these two approaches, as mentioned before. For the remainder of this thesis, the aligned terms will be used interchangeably to simplify the work with both data quality dimensions as defined in the DQV vocabulary as well as with FAIR metrics.

DQV	FAIR
Category	/
Dimension	Principle
Metric	Indicator
Measurement	/
/	Practical Test

Table 2.10: Alignment of the terminology between DQV and FAIR.

CHAPTER 3

Requirements

In this chapter, we discuss the requirements of a service for the automated evaluation of DMPs and provide the necessary background for the later conceptual design of a solution. In this regard, we will present requirements using a methodology which is common for this kind of publication. This includes an analysis of stakeholders, followed by the listing of use cases, and based on these use cases more fine grained functional and nonfunctional requirements. The collected requirements represent an interpretation of proposals found in previous work on the topic of maDMPs [SJM$^+$18, MWN$^+$21, FBM21, MRGB17, SJMM17, CPB20, PHM23, MSMJ19, MSS$^+$23] and from discussions with the maDMP community, particularly in the scope of the OSTrails[1] project.

As outlined in the comprehensive overview of Software Requirements Engineering by Wiegers et al. [WB13], use cases specify how users and stakeholders interact with the system to accomplish their objectives. Functional requirements are derived from these use cases and provide a specification for the implementation. When testing the implemented solution, it can be used to judge whether the use cases were properly implemented. Non-functional requirements are not associated with the use cases and define how a system should behave. To provide a uniform notation for the presented requirements we make use of the MoSCoW notation which provides a prioritization method for the communication of requirements.

3.1 Roles of Stakeholders

Miksa et al. [MSMJ19] identified nine stakeholders (*Funder*, *Ethics Review*, *Legal Expert*, *Researcher*, *Publisher*, *Repository Operator*, *Infrastructure Provider*, *Research Support Staff*, *Institutional Administrator*) with a role in realizing the maDMP vision depicted in Figure 3.1. These stakeholders can be grouped into three different roles, *DMP Maintainer*,

[1] https://ostrails.eu/

Figure 3.1: maDMP target audience [MSMJ19].

Reviewer, and *Review Expert*. As maDMPs are intended to be used as a common standard to integrate different services into the RDM infrastructure, all stakeholders mentioned above could assume any of those roles at different stages of the RDM lifecycle depending on the specific use case.

3.1.1 DMP Maintainer

Stakeholders editing the information of a DMP can assume this role. This includes researchers creating the DMP and adding information throughout the research, and other stakeholders such as repository operators, infrastructure providers, and research support staff providing information about their actions. Although stakeholders who have this role do not necessarily have the goal of reviewing the content of a DMP, they still benefit from the guidance provided by quality indicators.

3.1.2 Reviewer

Reviewers are the main group of stakeholders that the framework addresses. As mentioned in Section 1.2, the motivation for this thesis comes from the lack of an automated approach for the review of maDMPs. Reviewers should be assisted in the review process with metrics that have been automatically produced by the solution.

3.1.3 Review System Facilitator

As mentioned in Section 1.2 the review of DMPs requires specific domain knowledge and expertise that cannot be expected to be continuously available throughout the entire life cycle of the research data for every stakeholder. The availability of different practices and standards, depending on the research discipline, adds additional complexity to the work of reviewers. Therefore, the purpose of the role of *Review Expert* in the context of the proposed solution is to provide expertise during the configuration and integration of the solution in the review process so that automated indicators can be created based on this knowledge and other stakeholders lacking specific domain knowledge can benefit from the information provided by those indicators.

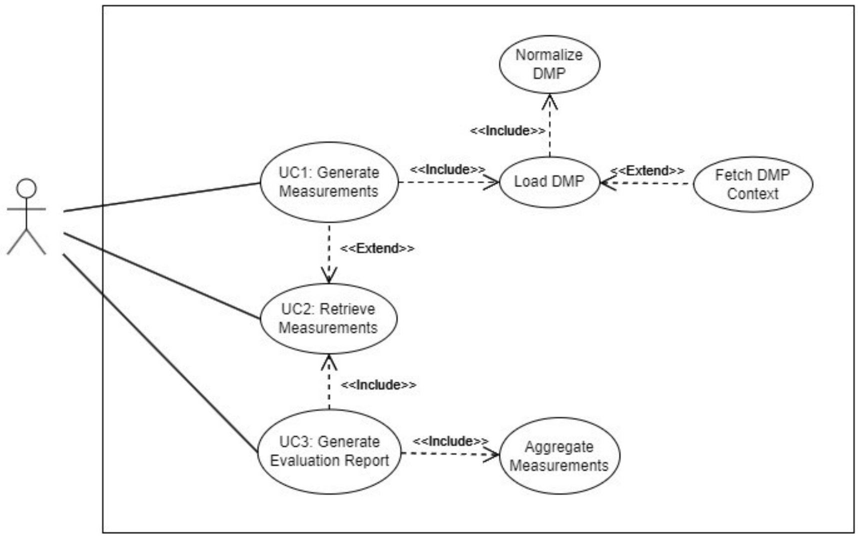

Figure 3.2: Use cases for the automated evaluation of DMPs.

3.2 Use Cases

Figure 3.2 shows the use cases of the proposed DMP evaluation solution that summarize propositions identified from literature and results of informal discussions with members of the OSTrails[2] consortium. Although in Section 3.1 we identified 3 stakeholder roles involved in the evaluation, all use cases of the proposed solution are relevant for any of these roles, as the evaluation of DMPs is a cross-cutting concern for all stakeholders involved with DMPs. There are three use cases that are relevant to an actor using the proposed solution:

1. **UC1 Generate Measurements:** Actors generate quality measurements for a given DMP which includes loading a maDMP from diverse sources and parsing it into a normalized format. In addition, information from linked resources and other data sources needed to generate measurements can be retrieved to provide context for the evaluation, which is an extension of the base use case for measurement generation. The process of generating DMP evaluation metrics depends on the specific evaluation evaluation requirements, and the framework must accommodate different scenarios based on the evaluation goals G1 - G4.

2. **UC2 Retrieve Measurements:** Actors retrieve the result of past evaluations without the need to recalculate the measurements. Measurements should be stored and made available to other actors and services.

3. **UC3 Generate Evaluation Report:** To assist reviewers in the evaluation of DMPs, the system provides the ability to combine measurements to provide a higher-

[2] https://ostrails.eu/

ID	Title	UC
FR1	Load maDMP	UC1
FR2	Fetch Context	UC1
FR3	Connect to SKGs and Repositories	UC1
FR4	Export Context	UC1
FR5	Produce Measurements	UC1
FR6	Connect to External Evaluators	UC1
FR7	Export Measurements	UC2
FR8	Export Evaluation Metadata	UC2
FR9	Generate Evaluation Report	UC3
FR10	Aggregate Measurements	UC3
FR11	Average Measurements	UC3

Table 3.1: Functional Requirements.

level evaluation result. The aggregation of measurements includes the selection of measurements based on a set of metrics, dimensions, or categories and the calculation of the sum and mean value of these measurements. In addition to this aggregation, the report contains other relevant information on the evaluation, such as the evaluated domain, the context harvested, and the measurements generated with their corresponding metric definitions.

3.3 Functional Requirements

Based on the definition of stakeholders outlined in Section 3.1 and the use cases for a DMP evaluation solution presented in Section 3.2, in this section we will elaborate on the functional requirements for the proposed solution to provide more precise requirements for the implementation of a solution. Figure 3.1 summarizes the functional requirements, which we will explain in the remainder of this section.

FR1 Load maDMP: The solution must be able to integrate arbitrary services that provide DMPs and load those maDMPs so that they can be accessed by different services.

FR2 Fetch Context: The solution must be able to extend the information of the given DMP with additional contextual data and store the result so that it can be accessed in the future.

FR3 Connect to SKGs and Repositories: The solution must provide the possibility to retrieve information from arbitrary SKGs and repositories independent of the underlying API, data model and technologies.

FR4 Connect to SKGs and Repositories: The solution must provide the possibility to access the context that has been collected for a DMP.

FR5 Produce Measurements: The solution must provide the possibility to produce quality measurements for DMP, given the maDMP to be evaluated, the data model underlying the maDMP to be evaluated, information about what quality dimensions should be included in the evaluation and contextual information for the given maDMP.

FR6 Connect to External Evaluators: The solution must enable the integration of external evaluation services such as FAIR evaluators independent of their implementation details.

FR7 Export Measurements: The solution must be able to store the quality measurements and provide access so that other services can access them after the evaluation.

FR8 Export Evaluation Metadata: Not only is the result of the application of a quality indicator of interest, but more information is needed to qualitatively represent the evaluation output. Therefore, the solution must be able to provide access to further evaluation metadata and provenance information.

FR9 Generate Evaluation Report: The system must provide the possibility to generate evaluation reports from already existing measurements from a previous execution of the DMP evaluation. These reports should collect all information resulting from the evaluation, namely the evaluated DMP, the corresponding context, and the generated measurements.

FR10 Aggregate Measurements: The system must be able to provide aggregations on the generated measurements grouped by *Metrics*, *Dimensions* and *Categories* and make these aggregations accessible to other services.

FR11 Average Measurements: The system must be able to provide mean values of the generated measurements grouped by *Metrics*, *Dimensions* and *Categories* and make those calculated averages accessible to other services.

3.4 Non-Functional Requirements

In this section, we will define non-functional requirements for the proposed solution. As mentioned in Chapter 1 performance-related requirements are not of concern for the proposed solution. In addition to that, certain requirements must be met to allow for seamless integration of the evaluation solution in the RDM infrastructure.

Extensibility DMP requirements vary depending on the reviewer and as outlined in Section 2.5, depending on the concrete evaluation use case there exist different guidelines that DMPs must comply with. Therefore, the solution should be able to provide a mechanism so that the evaluation of metrics can be extended and adapted to the given review requirements.

Furthermore, the landscape of data registries and SKGs providing data useful for evaluation is vast, as outlined in Section 2.4 and it is not possible to assume the existence of a data source that provides a single source of truth for all the information required. Therefore, the solution should provide the possibility of integrating information from various sources to enable the generation of measurements.

Lastly, DCS is intended to cover the minimal requirements of a maDMP to provide interoperability between RDM systems. Stakeholders can provide extensions to meet their needs as outlined in Section 2.5. Taking into account the existence of arbitrary DCS extensions, the solution should provide indicators not only for the information contained in the DCS standard but for the entire given DMP, including corresponding extensions and, therefore, also needs to provide mechanisms to accommodate these DCS extensions.

Compatibility The DCS standard has been developed as a means to facilitate the exchange of DMPs between different systems in the research data infrastructure. To further preserve this aspect of interoperability, the artifacts produced by the solution, such as evaluation metrics, should be maintained in a format that allows them to be passed between systems together with their respective maDMPs.

3.5 Discussion

In this section we defined the requirements for a solution for the evaluation of DMPs. We started by discussing identified stakeholders given from the literature and assigned them roles in the context of the proposed solution. In addition, we outline the use cases of the proposed solution and refined them into functional and non-functional requirements.

As the proposed deals with automated evaluation and assessment of DMPs, automation bias could be a resulting issue that must be taken into account. Automation bias, as described by Goddard et al. [GRW12] refers to the tendency of humans to rely excessively on automated systems. Because of this trust in automated systems, bias can occur when humans ignore or undervalue their judgment or information from other sources than the automated system. Given that the solution proposed in this thesis aims to assist reviewers in evaluating DMPs by providing automatically generated indicators, it is necessary to mention how this system avoids the fallacy of automation bias.

The solution aims to provide indicators to reviewers so that they can make more informed decisions when evaluating DMPs. Therefore, the indicators themselves should not express opinionated values, but measurable facts and information relating the DMP to existing standards or guidelines.

Additionally, as indicators relate only to machine-actionable information, there is no ambiguity in the provided indicators. DMPs also contain non-machine-actionable information that provides valuable insights for reviewers. In this solution, whenever free-form text is encountered, only the ability to query it will be provided without further processing. The possibility to provide metrics based on free-form text has been identified as a possible future work in Section.

In summary, on the basis of the above arguments, the issue of automation bias will not be considered in this thesis. Future work, however, especially when working with free-form text, should verify if the arguments above are still applying and if not consider the effects of automation bias.

CHAPTER 4

Conceptual Design

In this chapter, a conceptual design of a software framework for the evaluation of DMPs will be presented that addresses the requirements outlined in Chapter 3.5. To this regard, different components and services will be presented and outlined how they are supposed to be integrated with each other and what standards and interfaces are necessary to facilitate the solution. To communicate the proposed conceptual design, we reuse existing methodologies to present a reference architecture using common enterprise and software architecture methodologies.

Giachetti et al. [Gia16] describe an Enterprise Architecture (EA) as a solution that satisfies the requirements of all stakeholders of an enterprise regarding its information technology architecture. An EA is a comprehensive strategy and integrated method that organizations use to efficiently synchronize their business processes, information systems, technology infrastructure and human resources with their overall goals and objectives.

Angelov et al. [AGG12] describe software reference architectures as standardized reusable frameworks that provide guidance and best practices for designing and implementing software systems within a particular domain or application area. These architectures offer a structured blueprint consisting of components, patterns, and relationships that represent the common characteristics and requirements of systems within that domain. A software reference architecture can be part of an overall EA to further provide a structure to describe the architecture of the included software components.

To describe the reference architecture proposed in this framework, we will make use of the TOGAF EA framework [1] published by OpenGroup [The09] to provide a structure for the description of the architecture. Furthermore, we will employ selected parts of the arc42 template, which is a framework for the description of software reference architecture, to structure application architecture as part of the information systems architecture.

[1] https://www.opengroup.org/togaf

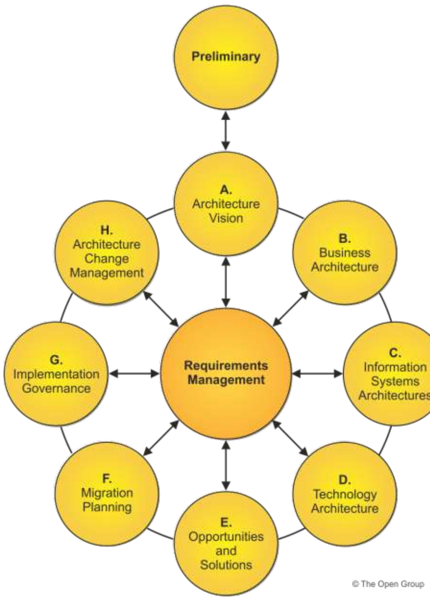

Figure 4.1: Structure of the TOGAF Architecture Development Method [The09].

4.0.1 Enterprise Architecture

Figure 4.1 pictures the Architecture Development Method which shows the cycles that TOGAF defines. These cycles provide a structured approach to the creation and management of enterprise architecture. As these cycles cover the entire process of an EA in an organization, it also includes cycles regarding evolution and management which are out of scope for this thesis. Therefore in this chapter we will only cover the following steps of the method:

- **B. Business Architecture:** Focuses on understanding and documenting key business processes. The resulting artifacts serve as a blueprint for aligning IT with business needs.

- **C. Information Systems Architecture:** Involves designing the architecture for individual application systems and includes the development of a data architecture and application architecture to support the business architecture where the *data architecture* defines the structure of the organization's data and the *application architecture* provides a blueprint for individual systems and the interaction between them, as well as the relationship with the business processes of the organization.

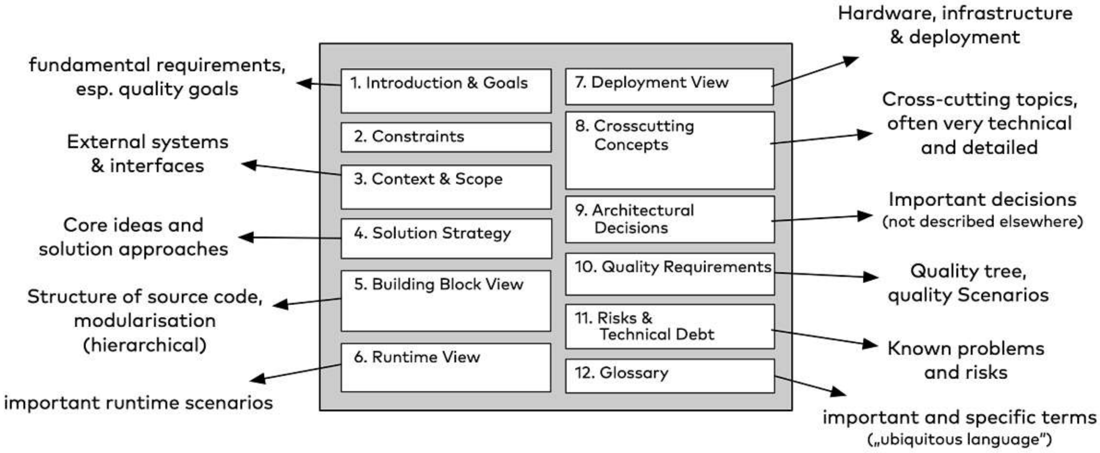

Figure 4.2: Structure of the arc42 Architecture Description Framework [GS22].

4.0.2 Software Reference Architecture Description

To describe the architecture of the proposed solution we utilize the software architecture description template provided by arc42[2] which provides a structure to document software architectures. Figure 4.2 summarizes the topics covered by the template, where for each topic, the template provides suggestions on what should be documented, as well as how the documentation can be communicated.

For this chapter only parts of the provided architecture description framework are required to describe the proposed solution because requirements and other constraints are already described in Chapter 3.5. In detail due to the scope of this thesis and this chapter, the following parts of the template will be utilized:

- **Context & Scope**: Consisting of *Business Context* which specifies all communication partners with explanations of domain-specific inputs and outputs or interfaces, and *Technical Context* to describe technical interfaces linking the system to its environment.
- **Building Block View**: Provides the static decomposition of the system into building blocks and helps maintain an overview of the technical aspects of the solution by making its structure understandable through abstraction.
- **Runtime View**: Documents the behavior and interaction of the system's building blocks and serves as a companion to the static building block view.

[2] https://arc42.org/

4.0.3 Modeling Language

A modeling language is a formalized way of representing systems, processes, or concepts using symbols, diagrams, or other graphical or textual notations. These languages are designed to facilitate the communication, analysis, design, and documentation of complex systems in various domains, such as software engineering, systems engineering, business processes, and more [Rum16]. The architecture frameworks mentioned in the methodology of this chapter are agnostic to the underlying details of representation and only provide guidance on the delivered artifact. To describe the included artifacts, we therefore make use of UML as a modeling language for software architecture descriptions as well as BPMN diagrams for the illustration of business processes.

4.1 Business Architecture

As part of step *B. Business Architecture* of the TOGAF EA architecture development method, in this section, we present the business workflows of the proposed framework using Business Process Model and Notation (BPMN)[3] which is a standard notation for describing business processes. These business processes are derived from the use cases defined in 3.2 and focus not only on the execution of the solution itself, but also highlight the processes involved in specifying and configuring these automated processes. Figure 4.3 shows a high-level overview of the business processes involved, which are:

1. **Evaluator Configuration**: Review experts configure the solution, providing domain knowledge that allows later measurement generation. This includes the definition of metrics and the evaluation context, as well as the configuration of evaluation methods.

2. **Evaluation**: The solution automatically generates indicators on a given DMP based on the configuration resulting from the process *Evaluator Configuration*.

3. **Evaluation Report**: To integrate the solution into the broader RDM infrastructure of an institution, the solution provides access to the generated indicators in a standardized format as well as methods to query individual measurements and aggregated information of these measurements.

These three processes are composed of further sub-processes, which will be explained in the remainder of this section.

Figure 4.3: Overview of processes of the proposed solution.

[3] https://www.omg.org/spec/BPMN/2.0/

4.1.1 Evaluator Configuration

The workflow of configuring the evaluation framework is pictured in Figure 4.4 and consists of the following steps to be carried out by the domain expert:

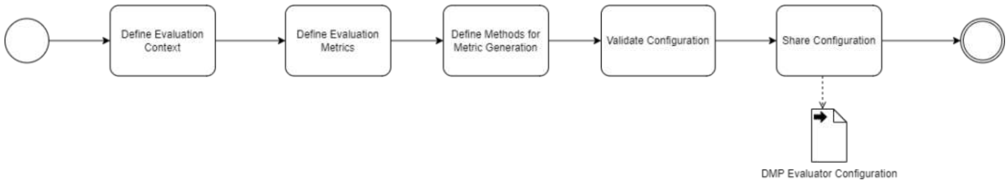

Figure 4.4: Evaluator configuration process.

1. **Define Evaluation Context**: DCS DMPs contain references to external resources that contain the mentioned information. In addition to explicit references to other data sources, the information contained in a DMP can be enriched with information from a variety of resources which have to be identified and mapped.

2. **Define Evaluation Metrics**: Different stakeholders have different evaluation and quality guidelines regarding DMPs and, therefore, no framework can produce all the indicators required for the use cases of each reviewer. Therefore, in this process, review experts define specific indicators and the respective unit of the resulting measurements to cover their needs. Furthermore, as outlined in Section 2.5 DMPs can be available at different stages of the life cycle of research data, containing different granularities of information, which also leads to the need to provide different metrics depending on the lifecycle phase. Therefore, when defining indicators, the lifecycle phase in which they can be measured should be taken into account.

3. **Configure Evaluation Methods**: After the form of the required indicators is defined, the semantics for creating measurements of them have to be specified. To this regard, domain experts provide definitions for evaluators with the goal of defining how quality measurements for the previously elicited metrics will be produced.

4. **Validate Configuration**: Before persisting the configuration, it must be validated itself to ensure that all the definitions in the configuration are consistent and sufficiently cover the evaluation requirements.

5. **Persist Configuration**: At the end of the configuration workflow, the resulting artifacts will be persisted in order to be used by the evaluation framework to provide the specified metrics in subsequent business processes.

4.1.2 Evaluation

Given a valid configuration, the evaluation process shown in Figure 4.5 generates and provides quality measurements for a given DMP. The following sub-processes are involved:

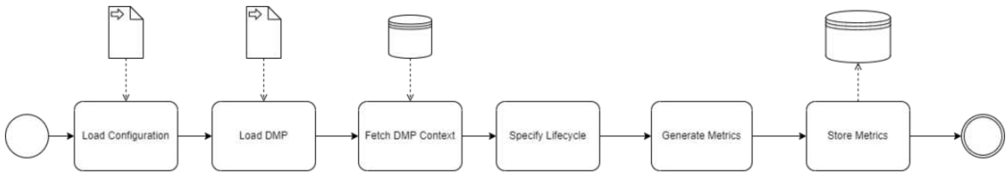

Figure 4.5: Evaluation process.

1. **Load Configuration**: The configuration exported as a result of the *Evaluator Configuration* workflow described in Section 4.1.1 will be instantiated and deployed as part of the evaluation solution.

2. **Load DMP**: The DMP for which indicators should be provided will be loaded into the environment.

3. **Fetch DMP Context**: The contextual information provided by the loaded configuration will be fetched according to the given configuration.

4. **Specify Lifecycle**: As outlined in the *Evaluator Configuration* workflow, indicators depend on the current lifecycle phase. Therefore, before generating any indicator measurements, reviewers must provide the DMP lifecycle phase to apply the given metrics.

5. **Evaluation**: This process produces the measurements for the given indicators according to the given configuration.

6. **Persist Measurements**: At the end of the evaluation process, the measurements will be stored together with the metadata of the corresponding metric.

4.1.3 DMP Evaluation Goals and Dimensions

Regarding the generated indicators, we propose a taxonomy of categories to dimensions as pictures in Figure 4.6. These are based on the high-level evaluation goals outlined in Section 1.1 where we listed some high-level goals regarding the evaluation of DMPs based on the proposal by Miksa et al. [MSS$^+$23] and Foidl et al. [FBM21] as motivation for the proposed solution. Starting with the goals *Completeness*, *Feasibility*, *Quality of Actions* and *Guideline Compliance* we refined them into review dimensions, referring to the quality dimensions described in Section 2.2 and FAIR principles presented in Section 2.1, and put them into the context of DMPs.

Although categories and dimensions can be deduced from previous work presented in Section 2.2, metrics on the other hand, are highly dependent on use cases and the given evaluation guidelines and requirements. Therefore, similar to the implementations of the FAIR principles we propose a set of metrics to cover the given categories and dimensions as a reference. Depending on the requirements at hand, these need to be adapted and extended to fit individual evaluation requirements. Additionally, the result of the evaluation given the same metrics is dependent on the implementation choices.

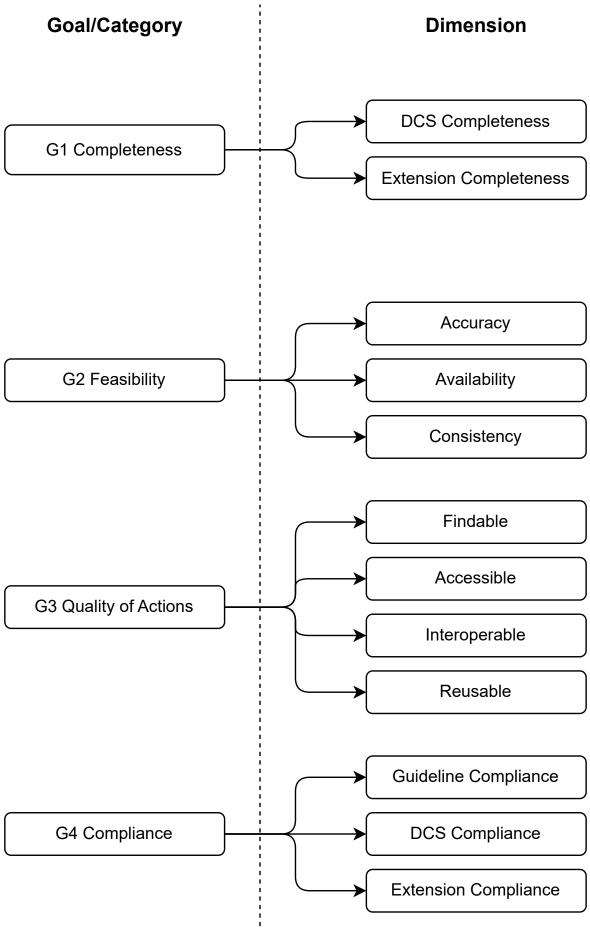

Figure 4.6: Categories and dimensions for maDMP evaluation.

Completeness In the scope of DCS DMPs, completeness can be further distinguished by the standard used as the baseline for completeness. The RDA proposed DCS as a minimal standard maDMP must comply to allow interoperability between RDM systems. Therefore, completeness with respect to this standard is a dedicated review dimension, namely *DCS Completeness*.

As DCS does not cover all aspects of individual stakeholders' needs, Cardoso et al. [CJM+20] have proposed extensions to increase the concepts a DCS DMP can cover. Therefore, the completeness of a DCS DMP with regard to arbitrary extensions is identified as a separate completeness dimension *Extension Completeness*.

Both completeness dimensions measure the extent to which necessary information is present in the maDMP and to provide measurements for these dimensions, we propose two metrics, *M1* and *M2*, listed in Table 4.1. These provide indicators on the existence of required entities and properties, respectively.

Id	Title	Description	Dimension
M1	Required Entity Existing	Indicate that a required entity defined in the schema is indeed present in the DMP	DCS Completeness, Extension Completeness
M2	Required Property Existing	Indicate that a required property defined in the schema is indeed present in the DMP	

Table 4.1: Completeness Metrics.

Id	Title	Description	Dimension
M3	Property Matches Ground Truth	Indicate that the value of a property is correct if there is some reference data that can be assumed to be true	Accuracy
M4	Existence of Linked Resource	Indicate that a linked resource is accessible, but not necessarily openly accessible	Availability
M5	Consistency Constraint	Indicate that the DMP is consistent with regard to a certain consistency constraint.	Consistency

Table 4.2: Feasibility Metrics.

Feasibility As described in Section 2.2, in the literature, this category is attributed to the three quality dimensions *Accuracy*, *Availability* and *Consistency*. For each of these dimensions, we propose Metrics as listed in Table 4.2.

Consistent with the definition given in Section 2.2, *Accuracy* refers to the degree to which the data values correctly represent real-world facts. In the context of maDMP evaluation, this implies comparing the values of the given maDMP with some ground truth. The result of this comparison is indicated by Metric *M3*.

Availability in the context of DMP evaluation is the degree to which a resource referenced in a DMP can be accessed. This is in contrast to the *Accessibility* principle of the FAIR principles [WDA+16], which provides information on how easy it is to access and use the data and focuses more on the available information describing accessibility properties of the digital object. To indicate the resulting accessibility of a resource referenced by a maDMP, we propose the Metric *M4*.

Lastly, *Consistency*, as outlined in Section 2.2 is concerned with the uniformity and reliability of the given data, while *Compliance* indicates if rules and standards are followed. We propose to measure the inconsistency with the Metric *M5* by indicating if the DMP fulfills a certain consistency constraint.

Quality of Actions In the data quality dimensions presented in Table 2.4 *Accessibility*, *Portability*, *Licensing* and *Compatibility* have been identified as dimensions of the category

Id	Title	Description	Dimension
M6	DMP Achieved FAIRness	Indicate the achieved value of a FAIR Metric on the published DMP digital object	Findable, Accessible, Interoperable, Reusable (FAIR)
M7	Dataset Achieved FAIRness	Indicate the achieved value of a FAIR metric of a published dataset object contained in the DMP	
M8	DMP Expected FAIRness	Indicate the expected value of a FAIR metric of a DMP as given by some guideline	
M9	Dataset Achieved FAIRness	Indicate the expected value of a FAIR metric of a dataset contained in the DMP as given by some guideline	

Table 4.3: Quality of Actions Metrics.

Quality of Actions. Because the dimensions included in this category are highly dependent on the applicable community standards, we opted to use the dimensions *Findable*, *Accessible*, *Interoperable*, and *Reusable* of the FAIR principles [WDA+16]. These principles are already widely used in the assessment of the quality of a wide range of digital objects, especially in the research domain. Therefore, there already exist different interpretations that address the needs of different scientific communities, as described in Section 2.1. As a maDMP references various digital objects for which FAIR measurements are already available, the use of metrics based on FAIR principles enables compatibility with other evaluation solutions.

Table 4.3 presents the types of metrics proposed for this quality category. When applying FAIR metrics on DCS DMPs we propose to distinguish between quality measurements on the DMP itself and measurements for the included datasets. Another distinction can be made whether the given resource is published or not. In case the DMP is published, actual FAIR metrics can be produced, and if not only predictions on the expected FAIRness and guidance can be inferred.

Based on these distinctions, *M6* indicates the achieved FAIR measurements on a published DMP and *M7* the achieved measurements on a published dataset referenced by the DMP. Furthermore, *M8* indicates the expected FAIR values of the DMP itself and *M9* of a dataset referenced in the DMP.

Compliance In the scope of DCS DMPs the *Compliance* of a DMP is dependent on the given requirements. Therefore, we identified three categories of regulatory sources that can provide compliance constraints for evaluation, which, in turn, constitute the dimensions of this evaluation category:

- **Guidelines:** Guidelines regarding the contents of DMPs are published by various entities such as funding bodies, universities and other stakeholders involved in the management of research data. An example of a funder guideline is the Science Europe Evaluation Rubric [Eur21], which provides the core requirements for DMPs, criteria for the selection of trustworthy repositories, and guidance for researchers to comply with organizational requirements.

- **DMP Common Standard (DCS):** maDMPs should follow a common data model to allow interoperability. As DMP Common Standard (DCS) is the maDMP standard endorsed by the RDA its constraints are compliance requirements for the evaluation of maDMPs.
- **DCS Extensions:** As described in Section 2.5.1, the authors of the DCS application profile for maDMPs recognized the need for extensions to the standard to cover application specific needs.

To measure the degree to which these dimensions are fulfilled by a given maDMP we propose 5 metrics, as listed in Table 4.4. *M10* indicates if a multiplicity constraint is fulfilled and an entity only has a defined number of properties of a type and likewise a defined number of references to other entities. *M11* indicates if an element of a maDMP is in a set of values that are not allowed and analogously, *M12* provides measurements that indicate if an element is in a set of allowed values. *M13* indicates if a value complies with a given pattern, so that, for example, email addresses have a correct format.

If a compliance constraint does not fall into one of the aforementioned metric categories, then it can be represented using *M14* which provides the result of a general verification of a guideline requirement. This, for example, can be the dependence of a value on the range of another value in a DMP or some other condition that can be automatically verified.

Non-ambiguity The review of non-ambiguity largely addresses parts of the DMP that are not machine-actionable and therefore also introduces a higher possibility of automation bias as outlined in Section 3.5. As only quality metrics relying on machine-actionable information will be considered in this thesis, the dimension *non-ambiguity* will not be included in the requirements but is listed as future work.

4.1.4 Evaluation Report

As a result of the business process *Evaluation* presented in Section 4.1.2 the solution can provide a set of indicators measuring different quality aspects of the DMP. The measurements alone, while containing all information resulting from the application of the evaluation solution, may not provide immediate benefits to the stakeholder because of their format, which is tailored to be consumed by machines. To provide a measurable result, these measurements can be aggregated to provide an *Evaluation Report* to reviewers. Figure 4.7 shows the business process associated with the provision of these reports, including the following steps:

Id	Title	Description	Dimension
M10	Multiplicity Constraint	Indicate that a certain multiplicity requirement given by a guideline is fulfilled	Guideline Compliance, DCS Compliance, Extension Compliance
M11	Blacklist Vocabulary Constraint	Indicate if the value of the property is not blacklisted	
M12	Whitelist Vocabulary Constraint	Indicate if the value of a property is in a list of allowed values	
M13	Value Pattern Constraint	Indicate that the value matches a given predefined pattern	
M14	Guideline Verification	Indicate that the DMP meets a requirement of a given guideline	

Table 4.4: Compliance Metrics.

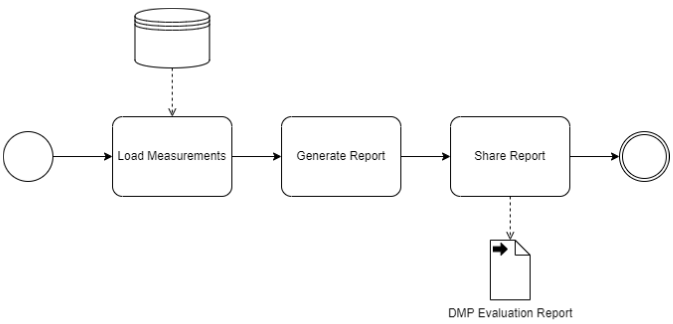

Figure 4.7: Evaluation Report Process.

1. **Load Measurements**: The quality measurement of an evaluated DMP will be fetched from storage.

2. **Generate Report**: A report will be generated by selecting measurement for certain metrics, dimensions, or categories and aggregating and averaging their values.

3. **Share Report**: Finally, the selected indicators together with the previously calculated aggregations and averages will be exported for further use.

Debattista et al. [DLA14] propose a solution to query aggregated metrics derived from base metrics which we adapt to fit the requirements of this solution. Based on the description of the evaluation result in the *Evaluation* business process presented in Section

4.1.2, the solution assigns individual measurements to metrics, dimensions and categories. In the following, we describe how the aggregations and averages for these aspects can be calculated to provide interpreted results v of the given quality measurements:

Measurement Assignment The function F produces the value of a metric m, which is either a number $\mathbb{N}$, a string $\mathbb{S}$ or a boolean value $\mathbb{B}$. String measurements are not suitable for further processing, but numbers and Boolean values will be included where Boolean values will be assigned to $false \rightarrow 0$ and $true \rightarrow 1$.

$$v : F_m \rightarrow \{\mathbb{N} \cup \mathbb{B}\}$$

Aggregation and Average of individual weighted Measurements The most fine-grained aggregation is over weighted individual measurements m, where each measurement m_i can be assigned a weight m_i, which is equal to $m = 1$ by default.

$$v(m_i) = w_i * v(m_i)$$

The sum of those weighted measurements $\sum_{i=1}^{n} v(m_i, w_i)$ determines a score that provides a measurable and comparable value across a range of metrics and can be used further to calculate the weighted arithmetic average $(\sum_{i=1}^{n} v(m_i, w_i))/n$.

Aggregation and Average of Measurements over a Dimension Similarly, as with metrics, reviewers might want to get a score in a less fine-grained manner by aggregating values over certain dimensions D while evenly applying weight m.

$$v(D, w) = \frac{\sum_{m \in D} v(m, w)}{|D|} = w * \frac{\sum_{m \in D} v(m)}{|D|}$$

Aggregation and Average of Measurements over a Category The highest level of abstraction is a *Category* C where each category can include multiple *Dimensions* D. Similarly to aggregating the metrics of a *Dimension*, given a weight w the weighed average $v(C, w)$ for *Measurements* in a *Category* can be calculated as follows:

$$v(C, w) = \frac{\sum_{D \in C} v(D, w)}{|C|}$$

4.2 Data Architecture

As part of step *C. Information Systems Architecture* of the TOGAF EA framework, in this section we will propose reference models for data structures relevant to the identified business processes, especially regarding communication with external partners and services. This includes data structures to communicate measurements, as well as methods to assign contextual information to a DMP. To represent the proposed conceptual data models, we will use UML class diagrams.

4.2.1 DMP Quality Vocabulary

The primary output of the evaluation of a maDMP in the scope of the solution proposed in this thesis are quality measurements representing instances of quality metrics. In this Section, we determine the requirements of a data structure describing the result of the evaluation of a DMP. To capture the requirements, we outline a set of competency questions listed in Table 4.5 that should be answered with the provided information. This includes information on the measurements themselves, the corresponding metadata to refer to the respective evaluated part of the maDMP, the associated metrics, dimensions and categories, and the provenance information from the evaluation.

Based on these competency questions, we modeled a data structure, shown in Figure 4.8 that provides answers to the given competency questions in order to provide a standard for storage and sharing of the results of the evaluation of a DMP. Table 4.5 additionally indicates which part of the model references which competency question. While designing this conceptual data structure, we focused on building upon existing standards. Therefore, the core of this model is based on the Data Quality Vocabulary (DQV) which is a W3C standard to describe data quality measurements. As such, it reuses definitions from other schemas such as Dublin Core (DC) and the PROV Vocabulary (PROV). We extended it with additional elements, to cover the specific requirements for representing DMP measurement results. Given that this data structure is an extension of DQV, we refer to the model as Data Management Plan Quality Vocabulary (DMPQV).

The core of the proposed data structure is a *Metric*, which provides information on what is being measured and how to categorize a measurement regarding its *Dimension* as well as *Category*. A *Metric* further provides information on the expected data type and value ranges. It could be that there exist metrics which, while measuring an aspect of the same dimension, form their own subgroup, and therefore these metrics can be assigned to a *MetricGroup*. Metrics specify a quality aspect of a DMP but can be a result of the application of individual tests, which are specified by a *MetricTestDefinitions*. A *Metric* is a template for a *QualityMeaurement*, which provides a value for the defined metric at a given *LifecycleStage* of a DMP and also provides a reference to individual *TestResult* instances. Quality Measurements point to the part of the maDMP which has been evaluated, which can be an entity or a specific value of an entity. Furthermore, Quality Measurements can refer to *Guidance* to provide suggestions for improvements, as well as to provenance information from the evaluation provided by *SoftwareAgent*.

Table 4.5: Competency questions for a vocabulary describing the results of a DMP evaluation.

ID	Competency Question	Term in DMPQV
1	What is the Dimension of a Metric?	Metric/inDimension
2	How can be dealt with more fine-grained definitions of dimensions?	Dimension/hasSubdimension
3	What is the Categoy of a Metric?	Dimension/inCategory
4	What is the coresponding metric definition of a measurement?	QuallityMeasurement/isMeasurementOf
5	How can metrics be uniquely identified?	Metric/identifier
6	If a Dimension contains metrics from different tools, how can they be distinguished?	Metric/inMetricGroup
7	What is the expected data type of a metric?	Metric/expectedDataType
8	What are the expected values of a metric and how should they be interpreted?	Metric/valueUpperBound Metric/valueLowerBound
9	At what stage of the data lifecycle can a metric be applied?	Metric/applicableLifecycle
10	If a metric is an aggregation and composed of individual tests, how can this dependency be described?	Metric/hasTestDefinition MetricTestDefinition/identifier
11	What is the expected data type of the individual tests?	MetricTestDefinition/expectedDataType
12	How should the value of an individual test be interpreted?	MetricTestDefinition/valueUpperBound MetricTestDefinition/valueLowerBound
13	What is the resulting value of a quality measurement?	QualityMeasurement/value
14	For what entity of a madmp has the measurement been calculated?	QualityMeasurement/computedOn DMPLocation/identifier DMPLocation/entity
15	Does this this measurement refer to a specific property of the entity for which it has been calculated?	DMPLocation/property

Continued on next page

Table 4.5: Competency questions for a vocabulary describing the results of a DMP evaluation. (Continued)

16	What is the corresponding metric of a measurement?	QualityMeasurement/isMeasurementOf
17	When was the measurement computed?	QualityMeasurement/generatedAtTime
18	At what stage of the data life cycle has the value been computed?	QualityMeasurement/appliedAtLifecycle
19	If there are recommendations on how to act on the result of the metric, what are they?	QualityMeasurement/hasGuidance
20	What agent generated the measurement?	QualityMeasurement/wasAttributedTo
21	If there are any associated test results of the measurement, what are they?	QualityMeasurement/hasTestResult
22	What is the value of a test result?	TestResult/value
23	What is the corresponding definition of a test result?	TestResult/hasTestDefinition

4.2.2 Evaluation Context

As DCS maDMPs contain references to external sources in the form of PIDs and other identifiers, we recognized the the need to resolve these references to provide an evaluation context for further evaluation of the maDMP, which has also been identified as a requirement of the proposed solution in Chapter 3.5. We propose to provide a standard representation of this context to complement DCS DMPs. To that extent, Table 4.6 represents the requirements of such a data structure in the form of competency questions that should be answered with the information contained in a data structure providing maDMP context. Figure 4.9 shows the proposed data structure that satisfies the competency questions, and the corresponding terms are also indicated in Table 4.6.

The main element of this conceptual component is *Context*. This conceptual element contains the data itself, which can be of arbitrary nature. To allow the interpretation by a consumer, a reference to the schema used is provided by the *Vocabulary* element. In addition, the context element also provides provenance information from which the information is gathered by referencing a *Source* component. Finally, a context has to be anchored with some entity or property of a maDMP which is made possible by linking to a *DMPLocation* component that holds the necessary information to address a DCS DMP.

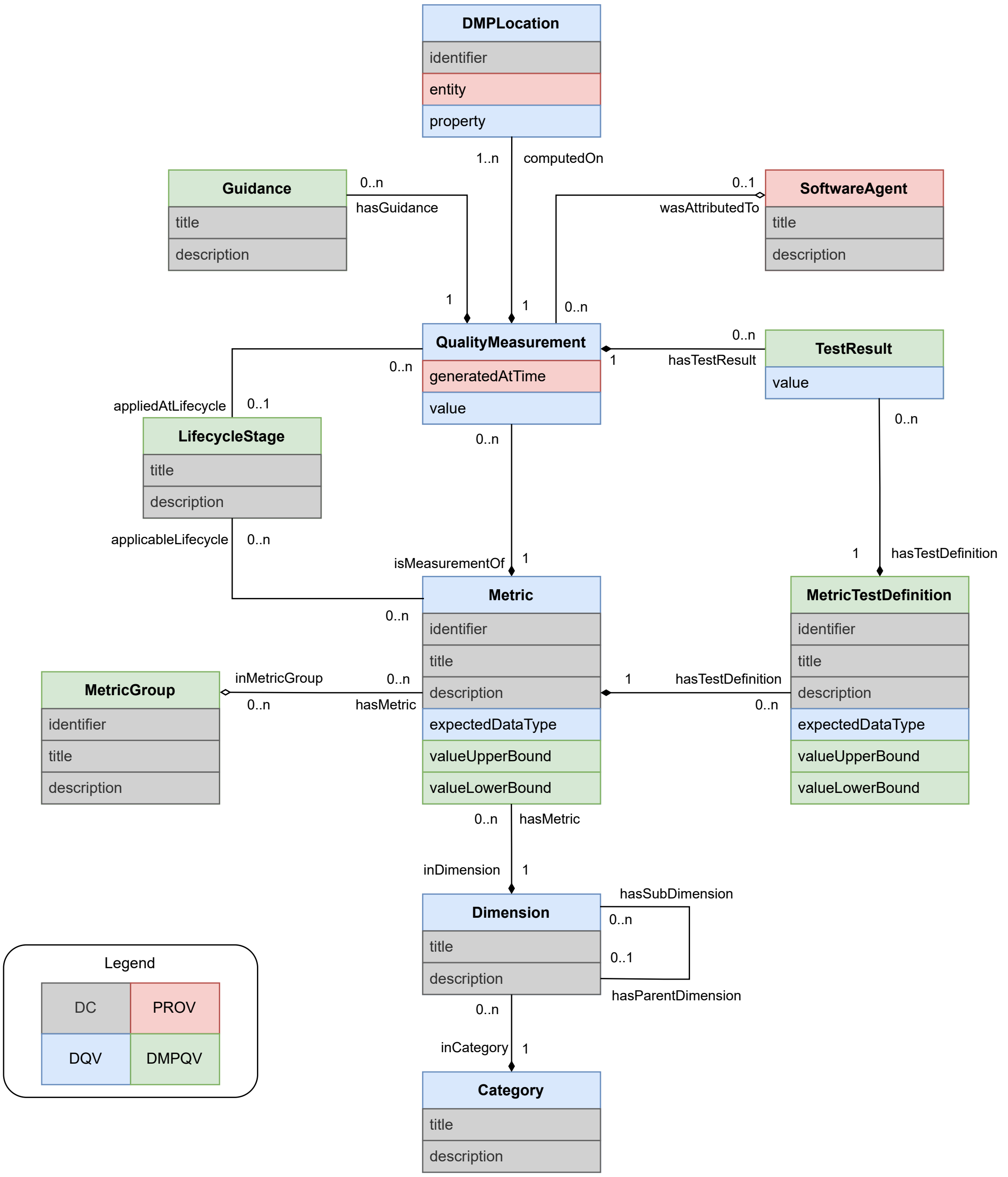

Figure 4.8: Depiction of the proposed Data Management Plan Quality Vocabulary (DMPQV) to describe DMP quality measurements based on the Data Quality Vocabulary (DQV) [AI20].

ID	Competency Question	Term in Context Vocabulary
1	What data is provided by the context?	Context/data
2	Where has the information been sourced from?	Context/hasSource
3	How is the data structured and what is contained in it?	context/hasVocabulary
4	To what part of the DMP does the context refer?	Context/hasDMPReference

Table 4.6: Competency questions for a vocabulary to describe contextual information of a DMP.

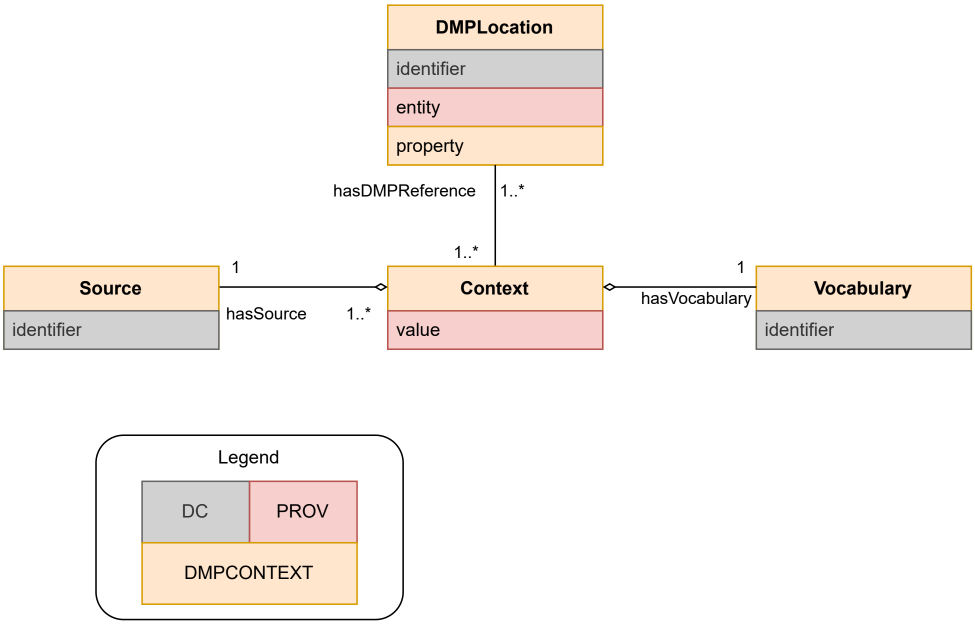

Figure 4.9: Depiction of the proposed vocabulary to describe DMP context.

To model the proposed DMP context model, we propose to reuse the existing vocabulary from the DC and PROV vocabularies as shown in the legend added to Figure 4.9. The choice of vocabulary used for the value of the context itself is left open by design, but when implementing a concrete context instance, we recommend using existing definitions such as those provided by schema.org[4] in the spirit of the FAIR principles to foster interoperability.

[4] `https://schema.org/`

4.3 Application Architecture

The second aspect defined in step *C. Information Systems Architecture* of the TOGAF EA framework is the description of an *Application Architecture*. To that extent, we chose to present this architecture using elements of the arc42 template for architecture communication and documentation. Following that template, in Section 4.3.1 we will provide an overview of all communication partners, in Section 4.3.2 we will provide a static decomposition of the system and lastly in Section 4.3.3 we describe the behavior of these building blocks as scenarios covering important use cases from the requirements given in Chapter 3.5.

4.3.1 Context and Scope

According to the arc42 documentation [GS22], system scope and context delimits the system from communication partners and therefore specifies the external interfaces. In this section, we provide business context by specifying the communication partners with explanations of domain-specific inputs and output interfaces with the goal of providing a common understanding of which data are exchanged with the environment of the system.

Table 4.7 provides an overview of the actors and systems with which the proposed solution will interact and provides a description of the expected exchanged information. These services are not part of the proposed software architecture itself, but part of the reference architecture for the evaluation DMPs.

4.3.2 Building Block View

To show the static decomposition of the system into building blocks and to make the structure of the proposed DMP evaluation framework understandable through abstraction, Figure 4.10 shows a high-level overview of the services included in the proposed system, as well as the connections between them. In the remainder of this section, we give an overview of the components of the proposed framework, including a description of their purpose, exposed interfaces, and interactions with external actors and services. Some components are composed of sub-components for which we also provide more detailed explanations.

DMP Loader

This component is responsible for connecting the DMP evaluation framework external *DMP Sources* and providing the information contained in these sources in the form of the DMP itself and the associate DCS ontology and ontologies of included extensions. In addition to providing access to this information, this component is also responsible for normalizing the DMP into the agreed DCS format. As a result, the component provides the normalized maDMP as well as the underlying DCS ontology and extension ontologies and abstracts the interaction with the *DMP Source*. To provide access to this collected information to other components in the framework, the *DMP Loader* exposes the

System	Description
DMP Source	Provides DMPs in a machine-actionable format.
SKG	Provide information on various research products, researchers, and organizations. The information is interconnected, and the entities contained can be referenced by a PID.
Data Repository	Provides different kinds of information to a given query.
Reviewer	Can be systems or human actors accessing the evaluation system to receive quality indicators for a specific DMP as well as reports containing aggregations of measurements.
External Evaluator	Provides evaluation results for different digital resources. Common evaluators include FAIR evaluators which take the identifier referring to the location of the object and return the corresponding measurements. The format of these measurements can differ depending on the evaluator used.

Table 4.7: Business Context.

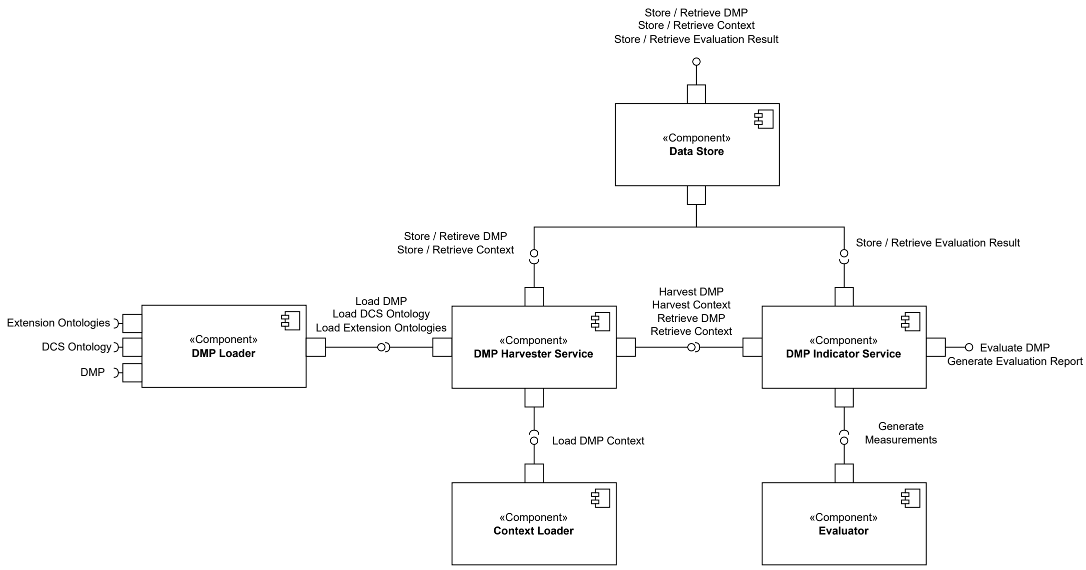

Figure 4.10: High-level building block view.

interfaces *Load DMP* to provide the maDMP for a given identifier, *Load DCS Ontology* to provide the underlying DCS ontology of the maDMP and *Load Extension Ontologies* to provide a list of extensions included in the DMP with the given identifier.

Context Loader

Provides contextual information for a given DMP by collecting and returning information from external sources such as *SKG*s and *Data Repositories* in a uniform format. We propose to make use of the proposed *Evaluation Context* data structure presented in Section 4.2.2 as part of the data architecture. The component exposes the interface *Load DMP Context*, which accepts a maDMP and returns a list of context objects that the *Context Loader* provides.

Data Store

The *Data Store* provides the ability to persist data and unify access to it in a way that is agnostic to the underlying format or technology used. In particular, it provides the ability to store and access maDMPs, the evaluation context, and the evaluation measurements resulting from the evaluation of a DMP.

DMP Harvester Service

This component unifies the access to data needed for the evaluation of maDMPs to provide a single endpoint to fetch the DMP itself including the associated ontologies and related contextual data, as well as to orchestrate the collection of this information from a variety of sources by accessing the interfaces provided by the *DMP Loader* and *Context Loader* components. To that extent, it exposes the interface *Harvest Extended DMP* which provides access to all this information.

Figure 4.11 shows a detailed view of the subcomponents contained in the *DMP Harvester Service*. It is made up of the components *DMP Provider*, *Context Provider*, *Inference Engine* and *Data Provider*.

DMP Provider The *DMP Provider* is responsible for loading normalized DMPs from relevant *DMP Loader* components. As actual retrieval of maDMPs from available sources and normalization to a uniform format is the responsibility of the individual *DMP Loader*, the *DMP Provider* acts as a bridge, connecting and exposing different instances of *DMP Loader* components with other components of the framework by exposing the *Load DMP* endpoint.

Context Provider The *Context Provider* component facilitates the communication between different instances of *Context Loader* components, which in turn collect and transform information regarding a given DMP from various sources such as SKGs and repositories. As a result, the *Context Provider* provides unified access to available instances of *Context Loader* components via the provided *Get Context* interface.

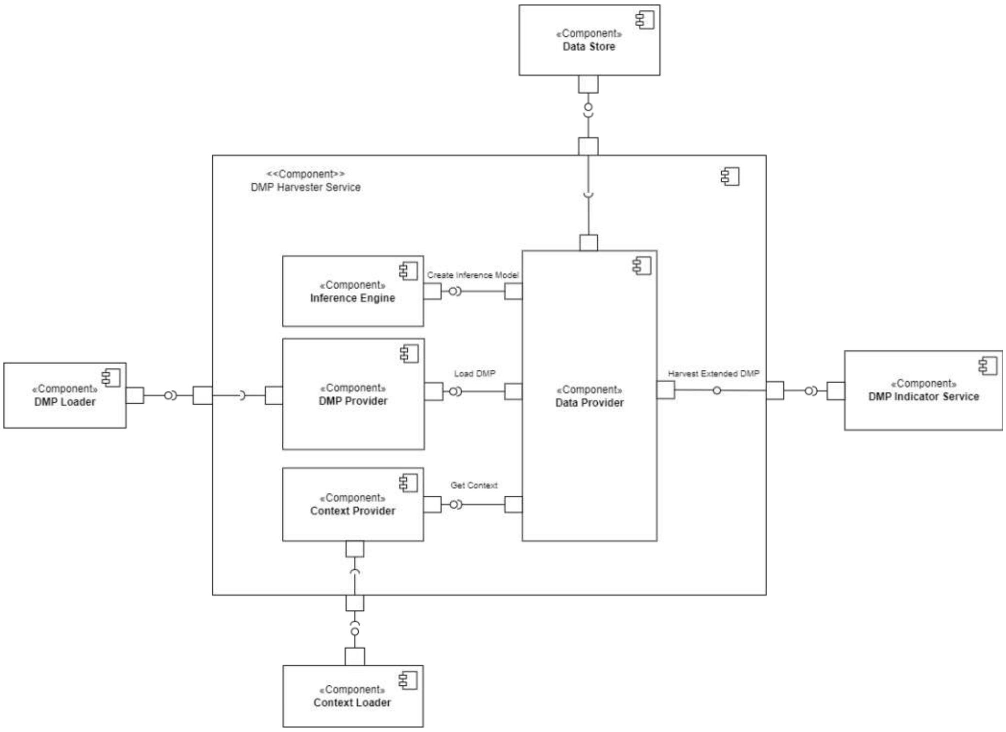

Figure 4.11: Building blocks of the DMP Harvester Service.

Inference Engine Linked data formats, such as RDF provide the ability to infer information which is implicitly contained in the data by deriving rules from corresponding ontologies and in turn provide the inferred information explicitly. The *Inference Engine* component abstracts this inference mechanism by exposing the *Create Inference Model* interface, so that other components can use the interface provided to derive new knowledge from their available data.

Data Provider The *Data Provider* collects information regarding DMPs from the *DMP Provider* and the corresponding contextual information from the *Context Provider*. It uses the interfaces of the *Inference Engine* component to expand the information that is implicitly available in the given DMP. Furthermore, the *Data Provider* component uses the *Data Store* component to store the DMP and the contextual information collected and provides the *Harvest Extended DMP* interface to export this information to other components of the framework.

Evaluator

The *Evaluator* component performs the actual calculations to generate quality metrics for a given DMP. In addition to a DMP that includes the associated ontologies of the underlying standard and included extensions, it can also receive evaluation context to have access to additional information related to the DMP. The way in which the evaluator generates metrics and what other external providers are needed for the evaluation is not defined by the reference architecture, as the specific requirements are dependent on the concrete evaluation requirements at hand and *Evaluator* components can also forward the generation of measurements to *External Evaluators* outside of the scope of the proposed system. To provide a clear separation of responsibilities, an *Evaluator* is supposed to provide evaluation measurements for only one evaluation dimension. The entire process of measurement generation is exposed with the interface *Generate Measurements*.

DMP Indicator Service

This component coordinates the evaluation process for a DMP and exposes the interfaces *Evaluate DMP* for the process of the DMP evaluation and *Generate Evaluation Report* for the the process of generating measurement reports in the form of aggregations and averages of measurements. The indicator service component retrieves the DMP, ontologies and additional context from the *DMP Harvester Service* and stores the evaluation result using the data store component. It coordinates the invocation of *Evaluator* components and provides them with the data they need for the generation of indicators. The indicators collected from the individual *Evaluator* components are collected and persisted at the end of the evaluation process.

Figure 4.12 shows the conceptual building blocks of the proposed *DMP Indicator service*. The service is composed of the sub-components *Evaluation Manager*, *Evaluation Provider* and *Metric Aggregator*. The remainder of this Section provides an overview of these proposed components, their scope as well as their provided as well as consumed interfaces.

Evaluation Manager The *Evaluation Manager* component is the central access point to request and retrieve quality indicators for a DMP by providing the interface *Evaluate DMP* to start a DMP evaluation process and *Generate Evaluation Report* to provide summaries of quality measurements. The component facilitates the generation of quality measurements by managing the collection of the necessary information through the interfaces of the *DMP Harvester Service* and gathers the generated measurements from the *Evaluation Provider* to which it passes the maDMP and other information and options required for measurement generation. The *Evaluation Manager* further accesses the *Data Store* component to store and access the evaluation results.

Evaluation Provider The *Evaluation Provider* is responsible for the integration of *Evaluator* components and provides a unified gateway for the *DMP Indicator Service* to access these components by providing the *Get Measurements* interface. It invokes *Evaluator* components, based on the evaluation requirements, and returns a collective result that contains the individual quality measurements of the respective *Evaluator* components. We recommend using the DMPQV data structure for the communication of quality indicators and measurements between the components involved.

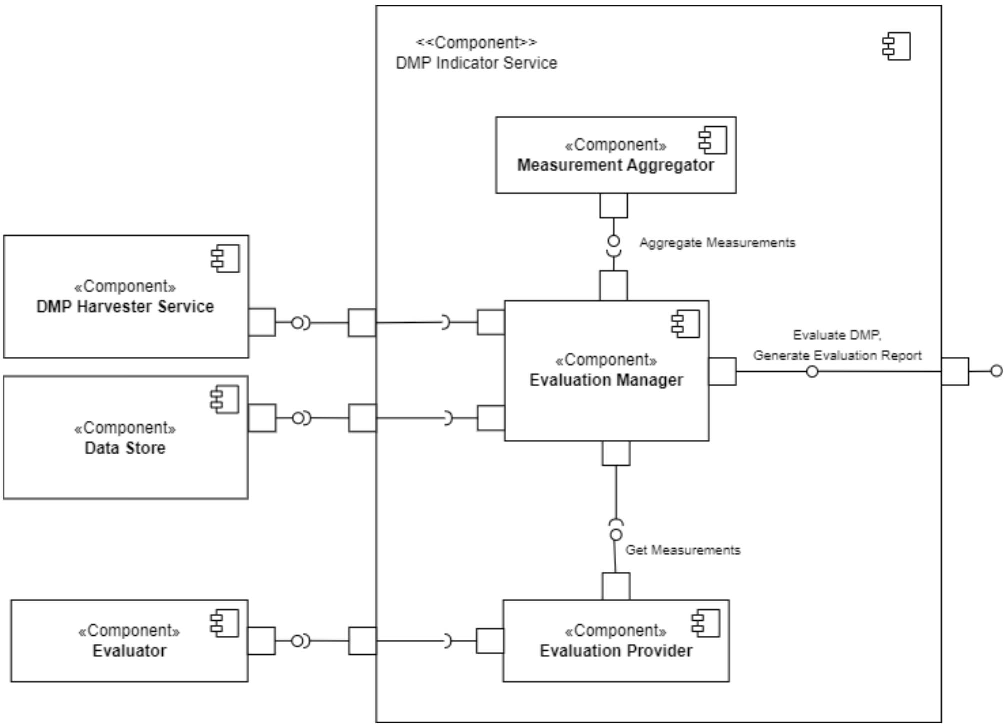

Figure 4.12: Building blocks of the DMP Indicator Service.

Measurement Aggregator The *Measurement Aggregator* exposes the interface *Aggregate Measurements* through which it retrieves quality indicators and processes them to provide a better view of the the result of the evaluation of a DMP by aggregating the measurements and calculating averages as described in the proposed of the business architecture.

4.3.3 Runtime View

The runtime describes concrete interactions between building blocks of the proposed system in order to provide and understanding how these building blocks communicate at runtime when performing certain use case scenarios. In this Section, we cover important use-case scenarios of the proposed solution based on the requirements elaborated in Chapter 3.5.

To this extent, we describe the scenario resulting from the use case *Generate Metric Measurements* which includes the use cases *Load DMP* and *Fetch DMP Context* and describe the interactions involved in the generation of DMP quality measurements including the retrieval of the underlying information in the form of maDMPs and supporting contextual information from external providers. Furthermore, we describe the interaction of building blocks from the point of the use-case *Create Evaluation Report* to highlight how the framework can further process the resulting DMP quality measurements.

Generate Measurements

Figure 4.13 shows the runtime view and interactions between the proposed components for the evaluation of DMPs with a focus on the interactions in the *DMP Indicator Service* component including the *Evaluation Manager* and *Evaluation Provider* components. Other components involved in the execution of this use case are the *Evaluator* component as well as the *DMP Harvester Service* and the *Data Store* component.

1. An *Actor* starts the evaluation process by accessing the interface provided by the *Evaluation Manager* which is a component of the *DMP Indicator Service*. With the request, the *Actor* provides the specification for the execution of the evaluation, such as details on which DMP to evaluate, as well as options regarding the evaluation process including details on which metrics should be considered for the evaluation.

2. The *Evaluation Manager* receives the request and calls the *DMP Harvester Service* to collect the necessary information needed for the evaluation. The *DMP Harvester Service* abstracts the access to this information and returns a maDMP in a normalized format together with the corresponding underlying ontologies, which describe the structure of the maDMP. Furthermore, it returns contextual information that augments the information present in the DMP.

3. The *Evaluation Manager* calls the *Evaluation Provider* and passes on the information previously received from the *DMP Harvester Service* to obtain evaluation measurements.

4. The *Evaluation Provider* gathers references to the *Evaluator* components needed to conduct the evaluation. The actual generation of measurements is carried out by instances of the *Evaluator* component, which provides implementations to evaluate specific sets of metrics.

5. The *Evaluation Provider* calls the previously identified *Evaluator* components in parallel and retrieves the returned measurements.

6. After each invocation of an *Evaluator*, the *Evaluation Provider* adds the resulting measurements to a collection and finally returns them to the *Evaluation Manager* after all *Evaluator* instances have returned their results.

7. The *Evaluation Manager* calls the *Data Store* to save the resulting quality measurements and afterwards returns these measurements.

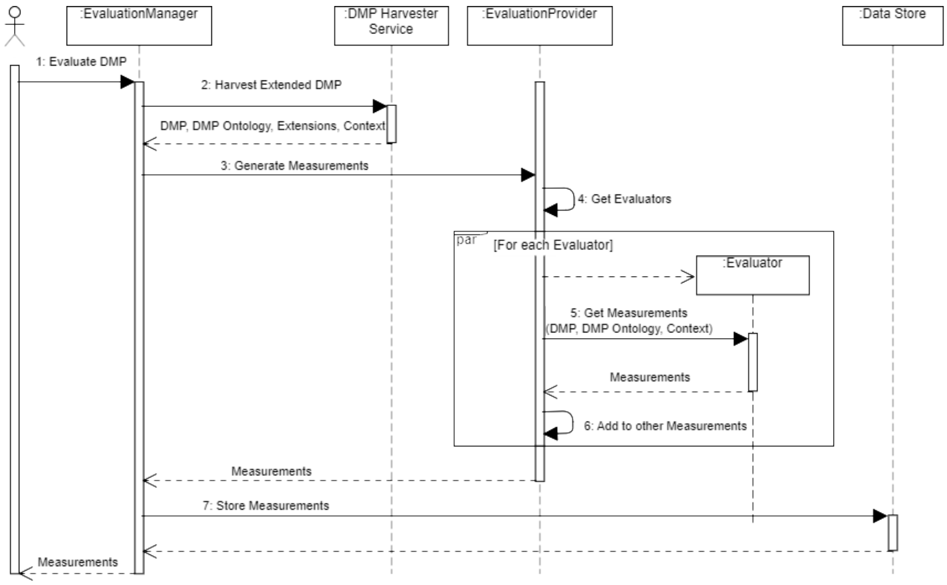

Figure 4.13: Runtime View: Generate Measurements.

Load DMP

Figure 4.14 depicts the interaction between the proposed components to implement the use case of retrieving maDMPs from various sources in an agreed upon format in order to further process them in the scope of the proposed DMP evaluation framework. The components involved in this scenario are parts of the *DMP Harvester Service*, more specifically the *Data Provider* component, the *DMP Provider* component and the *Inference Engine* component. In addition to the components of the *DMP Harvester Service* the component *DMP Loader* and the *Triple Store* are participating to fulfill the use case.

1. An *Actor* calls the *Data Provider* requesting to load a DMP for a given identifier.
2. The *Data Provider* calls the *DMP Provider*, which is responsible for managing the access to instances of *DMP Loader* components.
3. The *DMP Provider* accepts the request and selects an instance of a *DMP Loader* which can provide the DMP with the given identifier.
4. The *DMP Provider* calls the identified *DMP Loader* in order to retrieve the DMP as well as the definition of the underlying ontologies and extensions of the returned DMP and returns the retrieved data to the *Data Provider*.
5. The *Data Provider* calls the *Inference Engine* with the DMP as well as the underlying ontologies as parameters to derive additional knowledge and conclusions for the given DMP

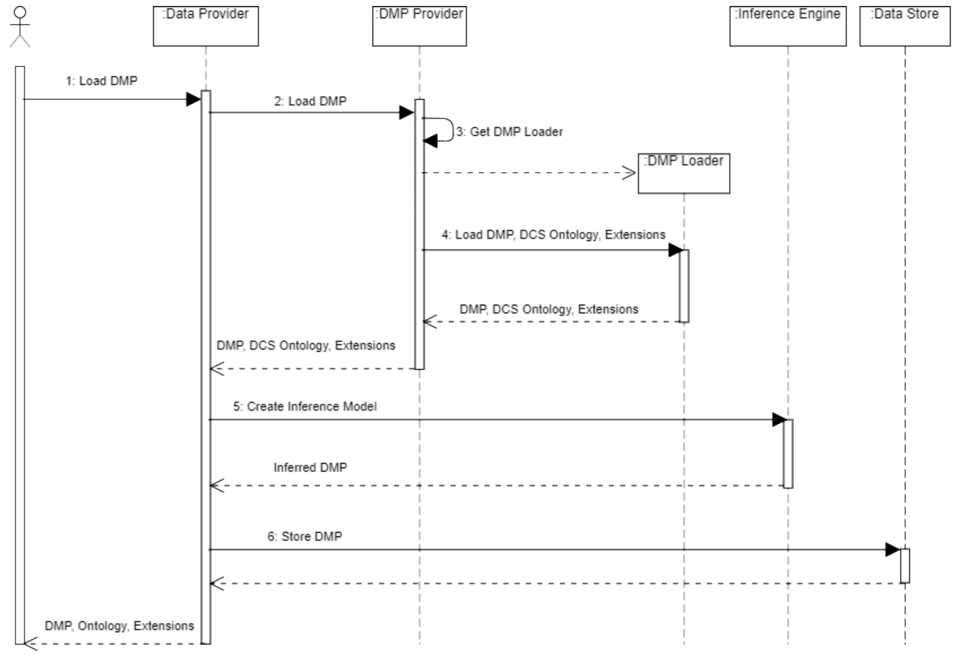

Figure 4.14: Runtime View: Load DMP.

6. In the last step the *Data Provider* stores the resulting inferred DMP and its associated ontology by calling the *Data Store* and afterwards returns the results to the *actor* thereby finalizing this process.

4.3.4 Fetch DMP Context

As part of the process of generating quality indicators, this process, which is depicted in Figure 4.15, collects relevant context to provide an extended view of the information contained in a DMP. To represent the context, we recommend making use of the corresponding data structure proposed in Section 4.2.2. The components involved in the retrieval of this context include parts of the *DMP Harvester Service*, more specifically the *Data Provider* component as well as the *Context Provider* component. Other components involved are the *Triple Store* to persist the resulting context as well as *Context Loader* components that abstract the connection to external information providers.

1. An *Actor* requests contextual information for a certain DMP from *DMP Harvester Service* which forwards the request to the Data Provider component.

2. The *Data Provider* receives the request and calls the *Context Provider* with the given DMP as a parameter to further handle the retrieval from different external resource providers.

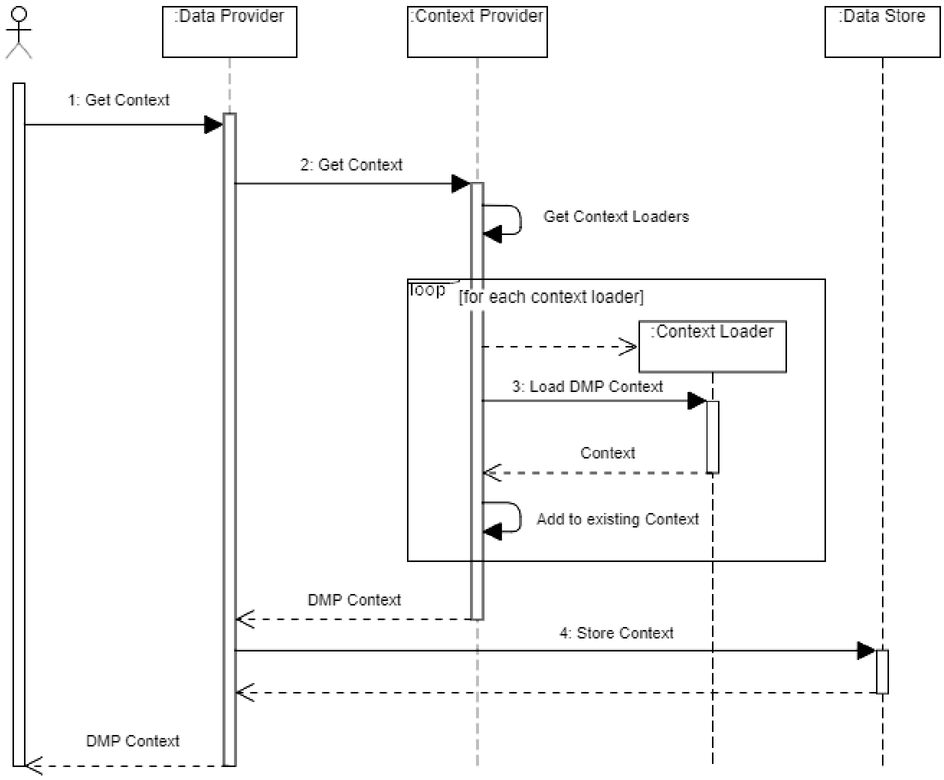

Figure 4.15: Runtime View: Fetch DMP Context.

3. The *Context Provider* collects the available instances of *Context Loader* components and iterates through them in parallel. For each *Context Loader* instance it receives the provided information. While the *Context Loader* instances fetch this information from different resources, they should return the data in a uniform format. After the *Context Provider* collected the result from all *Context Loader* instances, it collectively returns the results to the *Data Provider*.

4. The *Data Provider* calls the *Data Store* to persist the resulting DMP evaluation context and returns the results to the actor initiating this process.

Create Evaluation Report

The scenario of generating evaluation reports through the aggregation of values from measurements of an evaluated DMP is shown in Figure 4.16. It requires communication between the components of the *DMP Indicator Service*, namely the *Metric Aggregator* component and the *Evaluation Manager* as well as the *Data Store*.

1. An *Actor* calls the exposed endpoint of the *Evaluation Manager* and requests a report for a DMP for which measurements have already been generated.

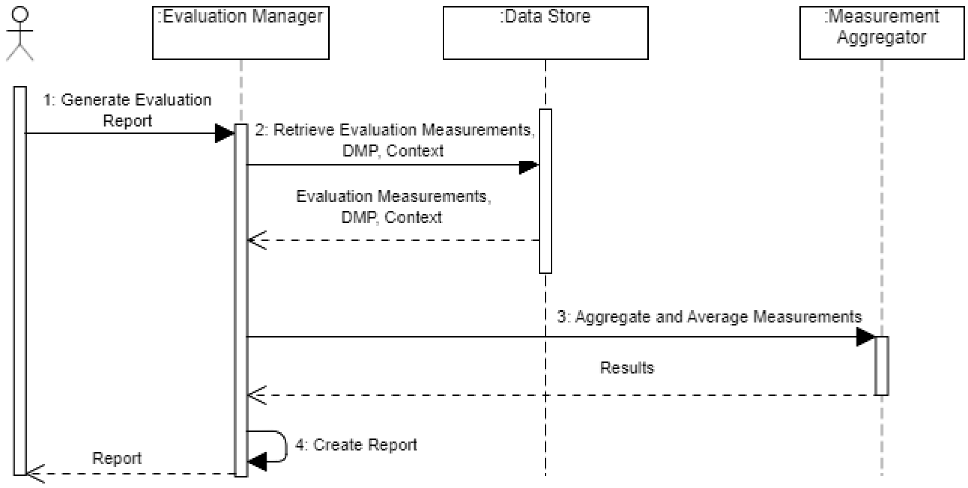

Figure 4.16: Runtime View: Create Evaluation Report.

2. The *Evaluation Manager* requests the corresponding measurements for the given DMP from the *Data Store* together with the associated DMP and context.

3. The *Evaluation Manager* calls the *Metric Aggregator* in order to produce the required aggregations in the form of sums of values and arithmetic averages of values for a selection of quality measurements.

4. With these aggregated values, the *Evaluation Manager* creates the evaluation report by combining the available artifacts and returns this report.

CHAPTER 5

Implementation

In this chapter, we implement a prototype for the evaluation of DMP by instantiating the conceptual architecture proposed in Chapter 4.3.4. As a context for the implementation, we present a case study as a specific setting in which automated evaluation is useful. This case study is concerned with evaluating the requirements outlined in the DMP evaluation guidelines of Science Europe [Eur21]. Furthermore we address the quality indicators proposed as part of the reference architecture.

5.1 Case Study

In this section we describe the context of our implementation of the proposed DMP evaluation framework. We build on the use case provided in the motivation of this thesis and aim to provide automated indicators for a DMP when it is submitted to a funder in machine-actionable form at the end of the research data lifecycle where all actions described in the DMP have already been implemented.

To represent funder guidelines we choose the Science Europe DMP Evaluation Rubric [Eur21] which is based on the core requirements for DMPs of Science Europe and represents a consensus on the requirements for DMPs of different funding bodies in Europe. The evaluation rubric outlines the requirements that a DMP must meet by listing different criteria in textual form and corresponding performance levels, which indicate to what extent the criteria are met. These guidelines are well suited as a baseline for the evaluation of the proposed solution, as they represent a consensus on DMP evaluation between major funding bodies and are well defined. They are not adapted to the DCS standard for maDMPs and therefore during the course of the implementation a dedicated funder extension is necessary to cover all the required information in a DCS maDMP.

In addition to the provision of indicators measuring the compliance with funder requirements as part of the *Science Europe Guideline Compliance* dimensions, we provide measurements for the evaluation dimensions proposed in Section 4.3.4, being *DCS Completeness*, *Extension Completeness*, *Accuracy*, *Availability*, results of FAIR evaluation and *DCS Compliance* to cover the evaluation goals mentioned in the motivation of this thesis. The semantics of measurement generation for these dimensions can be deduced from the DCS application profile and the FAIR principles which are both accepted standards of their fields.

We do not address the dimension *Consistency* in the prototype due to the lack of a dedicated standard that outlines these requirements. But we assume, that *Consistency* is partly covered by the evaluation of compliance requirements and the assumption that the definition of the structure underlying the given DMP is consistent, or in other words, that the combination of DCS and added extensions is consistent.

5.1.1 Science Europe Funder Extension

As outlined in Sections 2.5.1 the DCS application profile has the intention of covering the minimal requirements a maDMP should comply with in order to provide a certain level of interoperability between RDM systems. In order to cover specific use cases the use of extensions has been proposed by Cardoso et al. [CCE+22] and with the publication of the DCSO serialization there is a proposal on a vocabulary to better cover requirements funders commonly have for DMPs.

To cover the evaluation of the DMP requirements given by the Science Europe DMP guidelines, we further extend the proposed funder extensions, described in Section 2.5.1, and introduce new entities and properties. Table 5.1 shows these additional properties *property_rights_explanation* and *format_justification* as properties of *distribution* and *data_recovery_explanation* as a property of *host*. To provide a flexibility, we choose to provide the possibility to link to an externally available resource through the use of identifiers, as is already common practice in the DCS data model.

5.1.2 Constraints

In this section we introduce constraints for the implementation which reflect certain limitations on the choice of underlying technologies, given by the research questions of this thesis.

DMPs in DCSO Format The RDA proposed the DMP Common Standard (DCS) as a common standard for the representation of maDMPs. The DCSO standard is an instance of the RDA recommendation representing the DCS application profile using an RDF OWL ontology. As it is an implementation of the DCS profile by the authors of DCS, we will use the DCSO format in the prototype. This also limits the technologies used when working with a maDMP to semantic web technologies suited for RDF data.

Parent Entity	New Property	Description	Type	#
distribution	property rights explanation	Provide information whether intellectual property rights are affected and if so, explain which and how will they be dealt with	property rights explanation	1
	format justification	Justify the use of certain file formats	format justification	1
property rights explanation	description	Free text to describe and explain affected property rights	String	1
	property rights explanation id	Identifier for a property rights explanation artifact	Id	0..1
format justification	description	Free text to justify and explain the use of certain file formats	String	1
	format justification id	Identifier for a format justification artifact	Id	0..1
host	data recovery explanation	Explain how the data will be recovered in the event of an incident	data recovery explanation	1
data recovery explanation	description	Free text to provide a data recovery explanation	String	1
	data recovery explanation id	Identifier for a data recovery explanation artifact	Id	0..1

Table 5.1: Additional properties to increase coverage of the funder extension with regard to the Science Europe DMP Evaluation Rubric.

Extensions as OWL Ontologies DMP extensions used in the prototype must be in the same format as the provided DCS DMP. As specified above, the format used to represent DMPs is DCSO. Therefore, the extensions considered in the proposed solution must also be OWL ontologies in RDF format and compliant to the DCSO DMP.

SHACL for constraint validation Given the underlying RDF structure of the data we will use SHACL, introduced in Section 2.3, to validate the structure and content of RDF data as it is a W3C recommendation [1].

SPARQL for querying the DMP Following the constraint of DMPs being in represented as RDF triples according to the DCSO ontology, the choice of query language to be used in the prototype must be able to work with these triples. As the W3C recommendation for a semantic query language is SPARQL [2] we will follow that recommendation and use it in the proposed solution.

5.2 DMP Quality Vocabulary

We implemented the data structure proposed in 4.3.4 as an OWL ontology which is referenced by the w3id PID `https://w3id.org/dmpqv`. The hosted ontology reuses existing ontologies as proposed in the conceptual architecture.

5.3 DMP Evaluation Service Prototype

To evaluate the solution we implemented the reference architecture presented in Chapter 4.3.4 in the context of the given case study as a Spring Boot[3] application written in Kotlin[4] and refer to it as *DMP Evaluation Service*. To work with linked data structures, we integrated Apache Jena[5], a free and open source Java framework to build Semantic Web and Linked Data applications. The interfaces exposed to external actors, such as DMP evaluation and retrieval of resulting measurements, are implemented as REST APIs. To document these APIs we further integrated Swagger[6] to automatically document the exposed interfaces and foster reuse of the provided implementation. The complete source code, together with additional documentation, is available on GitHub[7].

Although the implemented prototype runs in the context of one single spring application, we split the dependencies between different packages of the implementation into three modules: *core*, *case-study* and *sdk* as pictured in Figure 5.1. The *core* package contains

[1] `https://www.w3.org/TR/shacl/`
[2] `https://www.w3.org/TR/sparql11-query/`
[3] `https://spring.io/projects/spring-boot`
[4] `https://kotlinlang.org/`
[5] `https://jena.apache.org/`
[6] `https://swagger.io/`
[7] `https://github.com/larnhold/maDMP-Assesment`

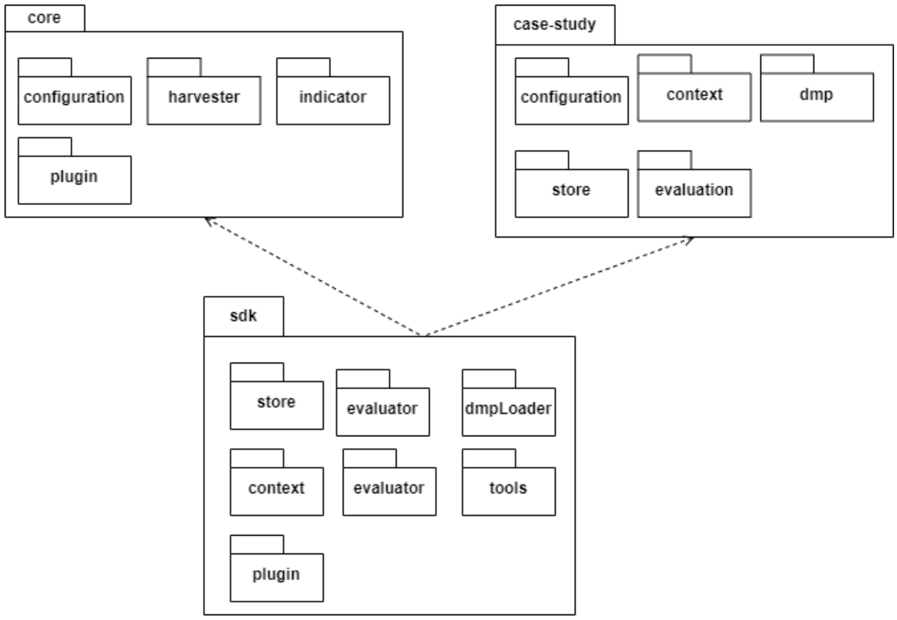

Figure 5.1: Package diagram of the implemented prototype.

the implementations of the *DMP Harvester Service* as well as the *DMP Indicator Service* which are parts of the proposed framework whose implementation is independent of the specific evaluation use case. The *case-study* package contains those use case dependent components such as instances of *DMP Loader*, *Context Loader*, *Data Store* and *Evaluator*. To provide a connection between components in these two packages, the *sdk* package provides necessary definitions of the interfaces and data structures to facilitate communication between components of the *core* and *case-study* packages.

Due to this separation, components of the universal *core* package do not have to directly reference a component from the application specific *case-study* package and vice versa the case-specific components of the *case-study* can can be written independently of the evaluation core by referencing the interfaces of the *sdk* module.

As the implementation of this thesis is based on the Spring framework, we make use of the built-in dependency injection mechanism to provide instances of components defined in the *sdk* package and implemented in the *case-study* package to the components and services of the *core* package. In addition to this default mechanism, we make use of the *Spring Plugin Project*[8], which provides the possibility of accessing multiple instances of the same component through the Spring application context. Therefore, in this reference implementation, the entire framework runs within the scope of one application to reduce the implementation overhead of a distributed system. It can be adapted with relatively little effort so that use-case-specific components such as *Evaluator* and *Context Loader* components can be hosted in a different environment and accessed by the core evaluation

[8] https://github.com/spring-projects/spring-plugin

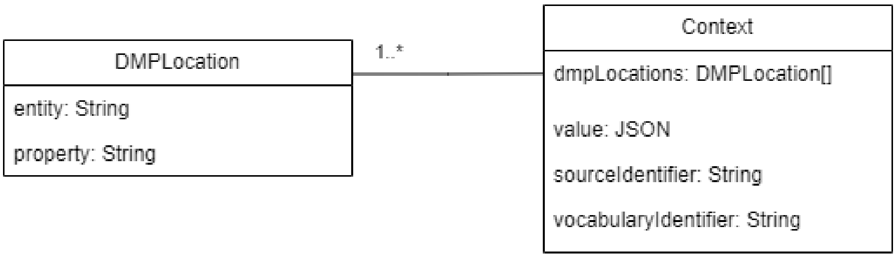

Figure 5.2: Instantiating the proposed DMP Context data structure.

services given the right resolution mechanism while the *Evaluator* and *Context Loader* components do not have to know the implementation details of other components other than the interfaces defined in the *sdk* package.

To enable interoperability between components of the implemented framework, we further integrated the standards to represent quality measurements and DMP context proposed in Section 4.2. As mentioned in the implementation constraints, when dealing with DMPs, a requirement is that they follow the DCS application profile in the form of the DCS ontology and therefore we represent DCS maDMPs using the DCSO serialization as an RDF graph. Extension mechanisms for DCS have been extensively discussed in the community and with the proposed RDF serialization a solution has been proposed on how to dynamically extend the information while still being able to distinguish the parts of a DMP referencing the DCS standard and parts that are added through extensions. This is achieved by using different namespaces for parts that describe content of the DCS application profile and parts that describe extension content.

To represent DMP context we integrate the corresponding data structure proposed in Section 4.2 into the solution as seen in Figure 5.2 and for the communication of the measurement results we also integrate the proposed DMPQV into the prototype. For simplicity, when implementing the storage functionalities of the *Data Store*, we opted to save the data to the file system in a serialized form. For future adoptions, this can be changed and more optimized storage technologies used.

Figure 5.3 shows the deployment of the proposed DMP evaluation prototype in the context of the given case study. *DMP Harvester Service* and *DMP Indicator* are implemented as proposed in Chapter 4.3.4 and facilitate the application and data flow as described in the corresponding activity diagrams. The key role of these components is to integrate other components of the framework in order to facilitate the use cases given in the requirements.

In contrast to these core components, the implementation details of the other components in the framework, namely *DMP Loader*, *Context Loader*, *Evaluator* and *Data Store* have been left open in the proposed conceptual architecture since the implementation requirements depend on the given scenario and the corresponding evaluation requirements. Table 5.2 lists the instances of these components that are implemented in the reference implementation and describes their responsibilities.

Component	Instance	Description
DMP Loader	JSON DMP Loader	In the scope of this implementation, we developed a DMP Loader instance which loads DMP from JSON files using the Jackson library [9]. Besides the DCS ontology this loader also parses information corresponding to the the proposed Science Europe Funder Extension.
Context Loader	OpenAire Context Loader	Resolves PIDs of Datasets contained in the DMP and receives corresponding information from the OpenAire SKG [10].
	Re3Data Context Loader	Resolves PIDs of hosts contained in the DMP and receives corresponding information from the Re3Data repository of research repositories [11].
Evaluator	DCS Completeness Evaluator	Measures the existence of all required properties a DCS DMP must contain.
	Science Europe Extension Completeness Evaluator	Measures the existence of all required properties with regard to the Science Europe Extension if the given DMP should comply with this extension.
	Accuracy Evaluator	Measures the accuracy of information contained in the DMP using the provided evaluation context.
	Availability Evaluator	Measures if provided references to external resources are existent and available.
	Achieved FAIRness Dataset Evaluator	Measures the achieved FAIR scores of the datasets contained in the DMP.
	Science Europe Guideline Compliance Evaluator	For each question contained in the Science Europe Evaluation Rubric, it is assessed whether the given DMP sufficiently addresses the requirement.
	DCS Compliance Evaluator	Measures whether the given DMP satisfies all structural requirements imposed by the DCS standard.
Data Store	Data Store	The implementation provided in the prototype provides the basic functionality of the component and stores data locally in the filesystem.

Table 5.2: Implemented components of the proposed framework to cover the requirements of the given case-study.

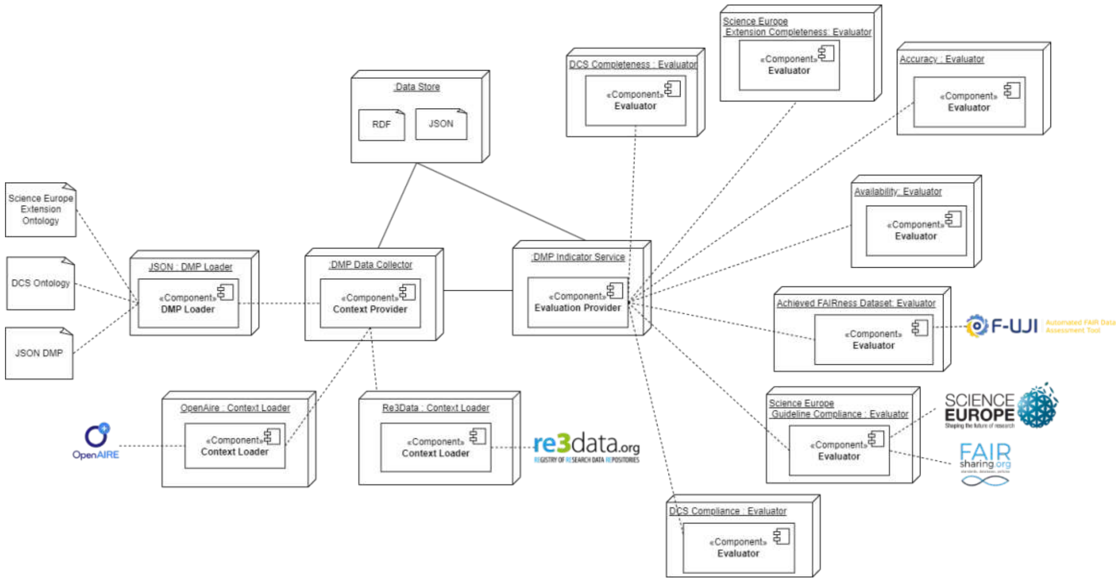

Figure 5.3: Deployment Diagram.

The remainder of this Section will further explain the implementation details of the *Context Loader* and *Evaluator* components implemented in the prototype.

5.3.1 Context Providers

With the use of the proposed format for the representation of DMP context it is possible to provide a static copy of the information provided by the resources referenced by identifiers contained in a DMP. This context, together with the DMP itself, provides a static snapshot of the information contained in a DMP so that reviewers and other stakeholders can access it. This is necessary as the information referenced by the identifiers could change, and therefore a persistent context provides a stable snapshot of this information. Figure 5.3 lists the entities in a DCS DMP that contain identifiers for external resources and the types of identifiers recommended in the DCS application profile. Especially when the type of an identifier is a PID the metadata of the referenced entity can be assumed to be accessible using the corresponding services of the provider of the PID.

For the scope of this thesis we limit the implementation to the inclusion of context for *Dataset* and *Host* entities by providing implementations of the *Context Loader* component of the reference architecture.

[9] https://github.com/FasterXML/jackson
[10] https://graph.openaire.eu/
[11] https://www.re3data.org/

DCS	Id	ID type
Contact	contact_id	orcid, isni, openid, other
Contributor	contributor_id	orcid, isni, openid, other
Dataset	dataset_id	handle, doi, ark, url, other
Distribution	access_url	URI
Distribution	download_url	URI
Distribution	license_ref	URI
Host	url	URI
Metadata	metadata_standard_id	url, other
DMP	dmp_id	handle, doi, ark, url, other
DMP	ethical_issues_report	URI
Funding	funder_id	fundref, url, other
Funding	grant_id	url, other

Table 5.3: Entities of the DCS [MWN+21] that contain identifiers to external resources.

OpenAire Context Loader

To gather additional information for datasets mentioned in a maDMP we query the API of the Openaire Graph which is one of the SKGs discussed in Section 2.4 provided by OpenAire, a pan-European research information system, which provides services to find, store, link and analyse research output from all disciplines[12]. Figure 5.4 shows the data model used in the OpenAire KG with following entities as described by Manghi et al. [MBA+]:

- **Data Sources:** The source from which metadata of an entity in the graph is collected from.
- **Projects:** Information on grants for funded research projects.
- **Research Results:** Represent the output of research activities, including publications, data, software, and other research products.
- **Communities:** Containing information on groups of people with a common research interest.
- **Organizations:** Correspond to companies or research institutions that are involved in projects or responsible for data sources.

[12] https://www.openaire.eu/

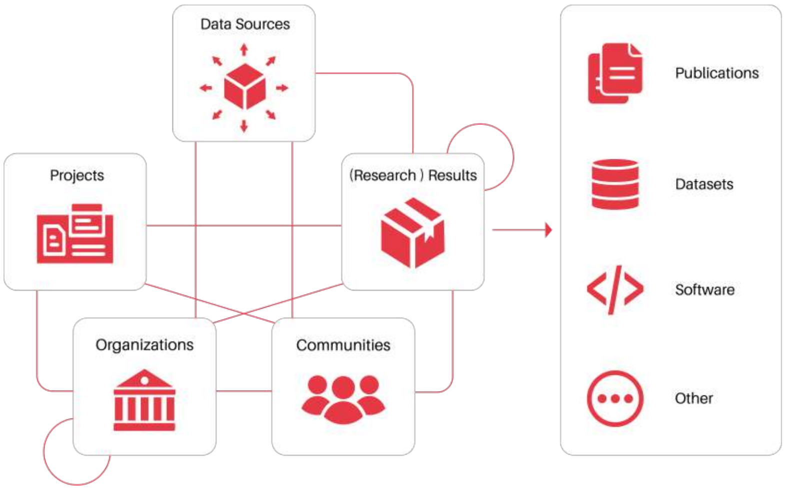

Figure 5.4: OpenAIRE Graph data-model [MBA+].

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dcso: <https://w3id.org/dcso/ns/core#>
select * where {
    ?dmp dcso:hasDataset ?dataset .
    ?dataset rdf:type dcso:Dataset .
    ?dataset dcso:hasDatasetId ?datasetId .
    ?datasetId dcso:identifier ?idValue .
    ?datasetId dcso:identifierType ?idType .
}
```

Listing 5.1: SPARQL Query to retrieve information on Datasets from a DCSO DMP.

The *OpenAire Context Loader* selects relevant values from the given DMP by running the SPARQL query outlined in Listing 5.1. This query returns the reference to the DMP, the reference to a dataset, and for each dataset the corresponding identifier entity with the identifier and the identifier type. With this identifier, the application queries the search API of the OpenAire graph and retrieves the available information.

The information retrieved is then serialized into a string and packed into a DMP context instance, pointing to the dataset and DMP retrieved by the previous SPARQL query.

Re3Data Context Loader

Re3Data is a global registry of research data repositories and has been in operation for more than 10 years. It provides a curated index of more than 3000 research data repositories from around the world and from all disciplines. The use of re3data is recommended in the European Commission's Guidelines on Open Access to Scientific

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dcso: <https://w3id.org/dcso/ns/core#>
PREFIX terms: <http://purl.org/dc/terms/>
select * where {
    ?dmp dcso:hasDataset ?dataset .
    ?dataset rdf:type dcso:Dataset .
    ?dataset dcso:hasDistribution ?distribution .
    ?distribution dcso:hasHost ?host .
    ?host terms:title ?title .
    ?host dcso:url ?url
}
```

Listing 5.2: SPARQL Query to retrieve information on Distributions and Hosts from a DCSO DMP.

Publications and Research Data in Horizon 2020. As an open science tool, it aims to promote data sharing, improve transparency, and support researchers in finding suitable repositories for depositing and accessing research data [GHG+13]. To that extent, we integrate the provided information through the implementation of the *Re3data Context Loader* to provide additional information regarding the hosts listed in the DMP.

The SPARQL query in Listing 5.2 queries the distributions and hosts associated with each dataset in a DMP. A *Dataset* in the DCS application profile follows the defintion of Dataset in the W3C DCAT specification and provides high-level information about the data. A *Distribution* in the DCS application profile is also synonymous with the term used in the W3C DCAT metadata application profile and is used to describe a particular instance of a dataset that has been or is intended to be made available in some way. Lastly, the host entity is intended to provide information regarding the system where data is stored and can be used to provide details on a repository where data is deposited.

Using the *title* and *url* provided in the host entity, we retrieve the corresponding entry from the Re3Data API which returns metadata for a repository in the form Metadata Schema for the Description of Research Data Repositories [SAB+23].

5.3.2 Evaluators

In this Section we will outline the implementation details of the Evaluators included in the prototype implementation. To give an overview, Table 5.4 lists the implemented instances of the *Evaluator* component and the corresponding evaluation scope. By design an *Evaluator* component provides metric measurements for one evaluation dimension that is indicated in the table, as well as the corresponding category of the dimension according to the DMP evaluation taxonomy proposed in Section 4.1.2. Furthermore, the table lists the corresponding types of metrics that the *Evaluator* components are considering in their evaluation, where the types of metrics are a subset of those proposed

Evaluator	Category	Dimension	Implemented Metric Types
DCS Completeness Evaluator	Completeness	DCS Completeness	M1, M2
Science Europe Extension Completeness Evaluator	Completeness	Extension Completeness	M1, M2
Accuracy Evaluator	Feasibility	Accuracy	M3
Availability Evaluator	Feasibility	Availability	M4
Achieved FAIRness Dataset Evaluator	Quality of Actions	FAIR	M7
Science Europe Guideline Compliance Evaluator	Compliance	Guideline Compliance	M14
DCS Compliance Evaluator	Compliance	DCS Compliance	M10, M12, 13

Table 5.4: Metric types implemented in the Evaluators of the Prototype.

as part of the business architecture in Section 4.1. Not all of the proposed types of metrics have been considered in the reference implementation, as in some cases there are no corresponding standardized guidelines to base the evaluation on, and the development of such guidelines is out of scope of this thesis. However, for all evaluation goals listed in the motivation of this thesis, we implement at least one evaluator to show the feasibility of the requirements.

Due to the underlying implementation constraint that DMPs are given as RDF triples, evaluators share common methods to query information and match patterns in the given DMP as part of evaluation processes. To query information, we use the methods contained in the *org.apache.jena.query* package which implements a SPARQL engine.

To verify if certain requirements of a DMP are met, such as the existence of certain properties, requirements on their values, and other structural requirements of the DMP which can be expressed with SHACL patterns, the evaluators make use of methods provided by Apache Jena in the *org.apache.jena.shacl* package. This package provides a runtime for SHACL constraints and returns a report in the form of the *SHACL Validation Report Vocabulary* indicating whether the data graph conforms to the shape graph and, if there are violations, contains entries that provide additional information. The content of the entries in this report is shown in Table 5.5 and parts of it can be used to create DMPQV measurements and metrics as indicated by the corresponding mapping in the table.

Property	Description	DMPQV Mapping
sh:focusNode	The focus node that was being validated when the error occurred.	DMPLocation/Entity
sh:resultPath	The path from the focus node. This property is optional and usually corresponds to the sh:path declaration of property shapes.	Measurement: Guidance
sh:value	The value that violated the constraint, when available.	DMPLoaction/Property
sh:sourceShape	The shape against which the focus node was validated when the constraint was violated.	
sh:sourceConstraint Component	The IRI that identifies the component that caused the violation.	
sh:detail	May point to further details on the cause of the error. This property can be used to report errors in nested nested shapes.	
sh:resultMessage	Textual details about the error. This message can be affected by the sh:message property.	Guidance
sh:resultSeverity	A value which is equal to the sh:severity value of the shape that caused the violation error. If the shape does not have the sh:severity declaration, then the default value will be sh:Violation.	

Table 5.5: Properties included in a SHACL validation result and mapping to the Data Management Plan Quality Vocabulary (DMPQV).

```
:DMPDatasetCompletenessShape a sh:NodeShape ;
    sh:targetClass dcso:DMP ;
    sh:property [
                    sh:path dcso:hasDataset ;
                    sh:minCount 1 ;
                ] .
```

Listing 5.3: SHACL Shape verifying that a DMP has at least one associated Dataset entity according to M1 Required Entity Existing.

In the remainder of this Section the implementation details of the *Evaluator* components included in the reference implementation are described, grouped by the evaluation category they address, which are equivalent to the review goals mentioned in the motivation of this thesis, namely *Completeness*, *Compliance*, *Feasibility* and *Quality of actions*.

Completeness Evaluation

The reference implementation provides two instances of *Evaluator* components addressing the *completeness* category. The *DCS Completeness Evaluator* verifies the existence of all relevant information as specified in the DCS application profile, while the *Science Europe Guideline Completeness Evaluator* assesses if all entities and fields required by the proposed Science Europe DMP Extension are included in the given DMP if the extension is specified as part of the given DMP.

To that extent, these two evaluator instances implement the recommended metric types *M1 Required Entity Existing* to evaluate how comprehensively maDMPs address various aspects of data management. Furthermore, *M2 Required Property Existing* verifies whether the DMP includes all mandatory fields required by the corresponding guideline.

In the reference implementation, both types of metrics are implemented using SHACL shapes verifying the existence of an element with the *minCount* property. Listing 5.3 verifies that a given DMP includes at least one *Dataset* as specified in the DCS application profile. The resulting evaluation report would produce metrics of type *M1 Required Entity Existing*. Similarly, the SHACL shape in Listing 5.4 verifies if a *Dataset* contains a property addressing the topic of sensitive data and therefore addresses the metric type *M2 Required Property Existing*.

By providing more shapes of this form, specifying the minimal counts for information required in the DCS application profile, we can produce measurements that indicate if the DMP is complete with regard to this standard. Similarly, completeness with regard to the proposed Science Europe funder extension is being assessed with SHACL shapes of this form.

```
:DatasetSensitiveDataCompletenessShape a sh:NodeShape ;
    sh:targetClass dcso:Dataset ;
    sh:property [
                    sh:path dcso:sensitiveData ;
                    sh:minCount 1 ;
                ] .
```

Listing 5.4: SHACL Shape verifying that a Dataset includes at least one field regarding sensitive data according to M2 Required Property Existing.

Compliance Evaluation

In the reference implementation we implemented the two evaluators *DCS Compliance Evaluator* to measure the compliance of a given DMP with regard to the DCS application profile and the *Science Europe Guideline Compliance Evaluator* to provide indicators targeting the requirements listed in the Science Europe DMP Evaluation Rubric.

DCS Compliance Evaluator In terms of compliance, the DCS application profile [MWN+21] defines requirements and recommendations on the form and content of the given maDMP, which we integrate into the implementation of this evaluator component.

One type of requirement given by the application profile concerns the multiplicity of entities and included values which aligns with the proposed metric type *M10 Multiplicity Constraint*. Compared to the completeness evaluation covered by the evaluator presented in Section 5.3.2, this evaluator evaluates multiplicity constraints more extensively in the form of minimal and maximum constraints modeled using SHACL patterns similarly to Listing 5.3 by specifying the properties *sh:minCount* and *sh:minCount* accordingly.

Furthermore, the DCS application profile provides recommendations on the values allowed for a certain property, as listed in Table 5.6. This compliance requirement is covered in the solution by providing metrics of type *M12 Whitelist Vocabulary Constraint*. To validate whitelist requirements, we again make use of SHACL shapes listing the allowed values with the *sh:in* property. Listing 5.5 contains a shape that defines the allowed values for the field *Sensitive Data* of a *Dataset*. By providing shapes of this kind, the whitelist requirements summarized in table 5.6 can be modeled.

Lastly, the DCS application provides some recommendations regarding the form of certain values such as date fields and URIs. As such data and time fields should be in a format compliant with ISO 8601 and the format of URIs is defined in RFC 3986. We propose to provide separate indicators covering these requirements with metrics of type *M13 Value Pattern Constraint*. Listing 5.6 shows how we implemented such a requirement of verifying the correct format of the issue date of a dataset with a SHACL shape including the definition of the corresponding regex pattern for ISO 8601 dates.

Data Property	Suggested Values
host:certified_with	din31644, dini-zertifikat, dsa, iso16363, iso16919, trac, wds, coretrustseal, none, other, unknown
distribution:data_access	open, shared, closed
dmp:ethical_issues_exist	yes, no, unknown
funding:funding_status	planned, applied, granted, rejected
distribution:format	MIME type (IANA media type)
host:support_versioning	yes, no, unknown
dataset:language	ISO 639-1 code
host:geo_location	ISO 3166 country code
contact_id:type	orcid, isni, openid, other
contributor_id:type	orcid, isni, openid, other
dataset_id:type	handle, doi, ark, url, other
dmp_id:type	handle, doi, ark, url, other
funder_id:type	fundref, url, other
grant_id:type	url, other
metadata_standard_id:type	url, other
host:pid_system	ark, arxiv, bibcode, doi, ean13, issn, handle, igsn, isbn, issn, istc, lissn, lsid, pmid, purl, upc, url, urn, none, unknown, other
funding:funding_status	planned, applied, granted, rejected
dataset:language	ISO 639-3 code
dmp:language	ISO 639-3 code
metadata:language	ISO 639-3 code
dataset:personal_data	yes, no, unknown
cost:currency_code	defined by ISO 4217
dataset:sensitive_data	yes, no, unknown

Table 5.6: Controlled Vocabulary recommended in the DCS Application Profile.

```
:DatasetSensitiveDataWhitelist a sh:NodeShape ;
    sh:targetClass dcso:Dataset ;
    sh:property [
                    sh:path dcso:sensitiveData ;
                    sh:in ("yes" "no" "unkown") ;
                ] .
```

Listing 5.5: SHACL Shape verifying that the value for the field *sensitive data* is included in the values recommended by the DCS application profile.

```
:DatasetDatePattern a sh:NodeShape ;
    sh:targetClass dcso:Dataset ;
    sh:property [
                    sh:path dcso:issued ;
                    sh:pattern "^[0-9]{4}-[0-9]{2}-[0-9]{2}
                    T[0-9]{2}:[0-9]{2}:[0-9]{2}(Z|[+-][0-9]
                    {2}:[0-9]{2})$"
                ] .
```

Listing 5.6: SHACL Shape verifying that the correct format of the *issued* field of a dataset with regard to the ISO 8601 specification.

1 DATA DESCRIPTION AND COLLECTION OR RE-USE OF EXISTING DATA			
Guidance for Researchers		Sufficiently Addressed The DMP...	Insufficiently Addressed The DMP...
1 a How will new data be collected or produced and/or how will existing data be re-used?	• Explain which methodologies or software will be used if new data are collected or produced. • State any constraints on re-use of existing data if there are any. • Explain how data provenance will be documented. • Briefly state the reasons if the re-use of any existing data sources has been considered but discarded.	• Gives clear details of where the existing data come from and how new data will be collected or produced. It clearly explains methods and software used. • Explains, if existing data are re-used, how these data will be accessed and any constraints on their re-use. • Explains clearly, if applicable, why new data must be collected, instead of re-using existing data.	• Provides little or no details on where the data come from and what data will be collected or re-used. • Does not, if applicable, provide sufficient rationale for generating new data.

Figure 5.5: Excerpt of die Science Europe DMP Evaluation Rubric [Eur21] specifying guidance for the necessary information to describe data collection and reuse.

Science Europe Guideline Compliance Evaluator Another type of guideline compliance we address in the prototype is the compliance with regard to the Science Europe DMP Evaluation rubric. This rubric contains guidelines for reviewers divided into six chapters covering different topics of DMP requirements and each chapter contains a set of relevant questions specifying DMP requirements. Besides guidance, the rubric also provides performance levels to indicate if the guidance is sufficiently addressed or not. Figure 5.5 shows a part of the Science Europe DMP Evaluation Rubric that gives guidelines on the required information regarding the collection and reuse of data.

To provide information on the compliance of the DMP with regard to these guidelines, we implemented the *Science Europe Guideline Compliance Evaluator* which provides measurements of type *M14 Guideline Verification* to indicate whether the given DMP complies with a guideline or not, thus matching the performance levels provided in the Science Europe Evaluation Rubric. The evaluator expects that the given DMP includes the *Science Europe Extension* proposed in this thesis, as otherwise the information required by some questions in the evaluation rubric cannot be queried from standard DCS DMPs.

To generate these guideline verification measurements, we build on the work of Foidl et al. [FBM21] who provide a set of SPARQL queries to retrieve the information required by the corresponding questions in the Evaluation Rubric and extended these queries with a mapping to the Science Europe Extension. Figure 5.7 shows a mapping of parts of the guidelines regarding data description and collection or re-use of existing data to the respective parts of the DCS application profile and Science Europe Funder Extension. As can be seen by the type of question they generally do not provide a standard to measure the validity of the extracted value, so in most cases the only kind of quality measurement that can provided includes information on the existence of the required information in the relevant mapped field. Therefore, we decided to use the same methodological approach as Foidl et al. [FBM21] and extract the values from a given DMP with SPARQL queries according to the mapping of the Science Europe Guidelines to the DCS application profile and the Science Europe Extension. Based on the result, we produce measurements indicating if all required values are present and if not provide the relevant information in the guidance of the measurement.

We identified three items of guidance that refer to the use of standards recognized by the community and listed them in Table 5.8. For these guidelines, we used the API provided by FAIRSharing[13], a FAIR-supporting resource that provides an informative and educational registry on data standards, databases, repositories, and policies, to provide measurements that indicate whether the values in DMP correspond to some community standard. That way part of the Science Europe evaluation rubric can indeed be automatically evaluated to some extent, given a measurable guideline and a trustworthy source of accepted standards.

Feasibility Evaluation

For the category of *Feasibility* we implemented the evaluator components *Accuracy Evaluator* providing measurements of type *M3 Property Matches Ground Truth* and *Availability Evaluator* providing measurements of type *M4 Existence of Linked Resource*.

Accuracy Evaluator The implementation of the *Accuracy Evaluator* produces measurement of type *M3 Property Matches Ground Truth* to indicate that information contained in the given DMP is valid given some reference information from a trusted

[13] https://fairsharing.org/

Guidance	DCS Mapping	Sciene Europe Extension Mapping
Explain which methodologies or software will be used if new data is collected or produced.	/	dataset/methodology
State any constraints on re-use of existing data if there are any.	distribution/data_access dmp/ethical_issues_exist	/
Explain how data provenance will be documented.	/	dataset/is_reused distribution/restriction_explanation
Briefly state the reasons if the re-use of any existing data sources has been considered but discarded.	/	dataset/methodology

Table 5.7: Mapping of selected guidelines for *Data Collection and Reuse* from the Science Europe DMP Evaluation Rubric to corresponding fields in the DCS Application Profile and the proposed Science Europe Funder Extension.

source. In the prototype, we assume that the contextual information provided to the evaluator when it is invoked is a trusted and truthful source. Additionally, the received context contains a pointer to the part of the DMP to which it refers. Given a mapping from a context to the DCS application profile, equivalences can be assigned and used to compare the values in the given DMP with the given context.

In Section 5.3.1 we describe the implementation of DMP context providers using the APIs of OpenAire[14] and Re3data[15]. These context providers provide contextual information for the entities *Dataset*, *Distribution* and *Host* contained in the DMP. Table 5.9 shows the mapping of the equivalent fields between the OpenAire data model and DCS application profile and Table 5.10 shows the respective mapping of equivalence between the Re3Data data model and the DCS application profile.

Based on these equivalence mappings, the implemented *Accuracy Evaluator* compares the values received through the context with the respective entries in the DMP and produces measurements that indicate if the value matches the evaluation context and if not, attaches guidance to the quality measurement to assist the reviewer.

[14] https://www.openaire.eu/
[15] https://www.re3data.org/

Category	Guideline	DCS Mapping	FAIRSharing Mapping
1b What data will be collected or produced?	Give preference to open and standard formats as they facilitate sharing and long-term re-use of data.	dataset/type distribution/format	Standard
2a What metadata and documentation will accompany the data?	Use community metadata standards where these are in place	dataset/metadata	Standard
3a How will data and metadata be stored and backed up during the research?	Give preference to the use of robust, managed storage with automatic backup, such as provided by IT support services of the home institution.	distribution/host	Database
5d How will the application of a unique and persistent identifier (such as a Digital Object Identifier (DOI)) to each data set be ensured?	Indicate whether a PID for the data will be pursued. Typically, a trustworthy, long-term repository will provide a persistent identifier.	dataset/dataset_id	Standard

Table 5.8: Alignment of Science Europe DMP Evaluation Guidelines to FAIRSharing entities.

Availability Dimension Evaluator This evaluator component measures the availability of digital resources referenced in a DMP and produces measurements of type *M4 Existence of Linked Resource*. The availability of a linked resource does not imply that it is openly accessible and such more sophisticated tests are already covered by various solutions in the field of FAIR evaluation. To verify availability, we implemented a simple ping of the referenced digital resource and return measurements depending on the received HTTP response code.

DCSO		OpenAire
Entitiy	**Property**	**Property**
Dataset	dataset_id	PID
	description	Description
	issued	Date of acceptance
	language	Language
	title	Title
	type	Type
	keyword	Subject
Distribution	data_access	Access Rights
	description	Description
	format	Format
	access_url	Publisher
	title	Title
	byte_size	Size
License	license_ref	License

Table 5.9: Mapping of fields from the OpenAire Data Model to the DCS application profile.

DCSO		Re3Data
Entitiy	**Property**	**Property**
Host	pid_system	pidSystem
	url	repositoryURL
	support_versioning	versioning
	certified_with	certificate
Dataset	metadata	metadataStandard
Distribution	data_access	dataAccess

Table 5.10: Mapping of fields from the ReData Data Model to the DCS application profile.

```
PREFIX dcso: <https://w3id.org/dcso/ns/core#>

select ?root ?id ?value ?type where {
    ?root dcso:hasId ?id .
    ?id dcso:identifier ?value .
    ?id dcso:identifierType ?type .
}
```

Listing 5.7: SPARQL Query to retrieve information on identifiers included in a DCS DMP.

This is rather straightforward for HTTP URLs as it is possible to select them based on their data type in the given RDF maDMP by querying all values that have the data type *xsd:anyURI*. However, there are also references in the form of PIDs and other identifiers contained in a given DMP that do not provide directly resolvable URLs. To that extent, the SPARQL query shown in Listing 5.7 queries all identifiers contained in a DCSO DMP and the corresponding type which identifies the PID provider that can resolve the identifier. Due to the underlying ontology, it is possible to cover all types of identifiers by selecting the entries according to their supertype *id*. In the prototype, we then provide availability measurements by accessing the APIs of DOI and ORCID resolvers depending on the type of the identifier.

Quality of Actions (FAIR) Evaluation

As highlighted in Section 4.1.2 the evaluation of FAIR metrics for a DMP can be manifold and we identified four cases for this kind of evaluation in the form of metric types *M6* to *M9*. These types consider the evaluation of different digital objects contained in a DMP at different stages of the DMP lifecycle. If the digital object has already been published, then it can be evaluated with regard to its FAIRness, otherwise only recommendations can be made for the reviewer to provide guidance throughout the stages of the research data lifecycle.

For the scope of this thesis we provide the *Achieved FAIRness Dataset Evaluator* to provide measurements regarding the achieved values of a FAIR evaluation for datasets referenced in the DMP that will be represented as metrics of type *M9 Dataset Achieved FAIRness*.

Achieved FAIRness Dataset Evaluator The FAIR evaluation of digital objects is a active area of research and as outlined in Section 2.1 there exists a wide variety of guidelines and tools to evaluate FAIR metrics. In the implementation of the *Achieved FAIRness Dataset Evaluator* we build upon this work and integrate the evaluation results

Figure 5.6: F-UJI evaluator result of dataset with DOI `10.5281/zenodo.4063720` [DH20].

of the F-UJI[16] FAIR evaluator. F-UJI exposes an API that takes a PID and evaluates the underlying digital object with regard to the FAIRsFAIR Data Object Assessment Metrics [DH21]. In addition to a JSON representation of the resulting measurements the F-UJI evaluator also provides a graphical representation of the achieved scores across the different FAIR dimension as seen in Figure 5.6.

To integrate the results of the F-UJI evaluator into the prototype, we added the docker image *ghcr.io/pangaea-data-publisher/fuji*. This service takes an identifier, extracted for each dataset with a SPARQL query, to a hosted dataset and returns the evaluation results and the results of individual tests in the format shown in Figure 5.7. In the prototype, we provide a mapping of this structure to the proposed DMPQV model to produce the respective quality measurements.

5.4 Discussion

In this section, we described the implementation details of a prototype of the proposed DMP evaluation framework based on a use case derived from the DMP evaluation goals proposed in this thesis and the Science Europe DMP Evaluation Rubric. The prototype serves as a proof-of-concept implementation to show the feasibility of the proposed solution for the evaluation of DMPs and to provide a basis for the evaluation of the conceptual architecture. For deployments in real-life scenarios, the implementation can be adapted depending on the given use-cases, and therefore the implementation is designed to be easily adapted and extended.

Furthermore, we proposed vocabularies and data models and provided reference implementations, but in the long term, we highlight the need for a wider range of standards in the ecosystem of maDMPs. The data models used to represent context in the reference implementation are modeled to match the information received from the information sources, namely OpenAire for information regarding datasets and Re3data for hosts, but follow no universal standards. In the context of the reference implementation, we can assume that all components can handle the vocabulary used since the components have been implemented jointly. The use of any arbitrary vocabulary for the representation of

[16] `https://www.f-uji.net/`

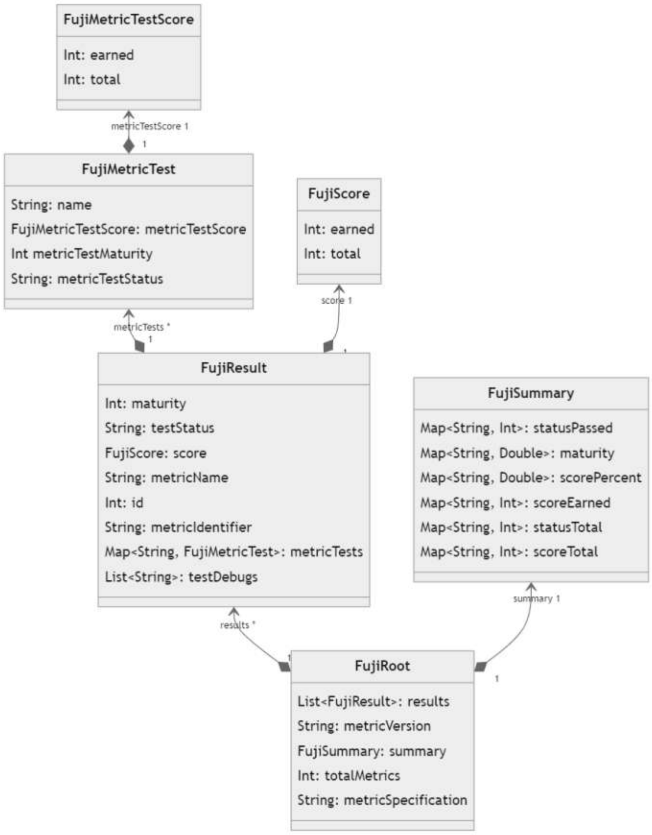

Figure 5.7: Structure of the evaluation result returned by the F-UJI [DH21] evaluator.

the context is sufficient for a reference implementation, but for an implementation on a larger scale it might be necessary to propose to reuse more standardized formats to represent the DMP context so as to enable interoperability between different systems. Some already existing standards such as the Datacite Metadata Schema[17], parts of Schema.org[18] or a subset of the vocabulary proposed in SKG-IF [19] could be reused for this purpose. The proposed data structure for the representation of the DMP context is flexible enough to accommodate different vocabularies in future implementations.

In addition to the proposed context format, the integration of Research Object Crates (RO-Crates)[20] can be considered to represent DMP context. RO-Crates is an approach to package research data with their metadata. These packages are based on schema.org annotations in JSON-LD and aim to make best practice in formal metadata description accessible and practical for use in a variety of situations [SRSC+22].

[17]https://schema.datacite.org/
[18]https://schema.org/
[19]https://skg-if.readthedocs.io/
[20]https://www.researchobject.org/ro-crate/examples

CHAPTER 6

Evaluation

In this chapter, we evaluate the proposed solution for the evaluation of DMPs based on the case study presented in Chapter 5.4 and the implemented prototype. To show the efficacy of the prototype, we provide a set of functional system tests that reference the use cases and functional requirements outlined in Section 3.3 and show that the requirements are covered.

Furthermore, given the evaluation goals and dimensions proposed in this thesis, we provide examples of manual evaluation of certain metrics for a minimal DCS compliant maDMP and compare the results of the manual evaluation with the corresponding quality measurements produced by the prototype to show the accuracy of the measurements and how these indicators can help reviewers. Additionally, we introduce some faults to this minimal DMP and verify if the solution properly recognizes and expresses these errors. Furthermore, we show to what extent the implemented quality indicators cover the given evaluation goals and highlight common evaluation methods to show the suitability of certain technologies for the provision of DMP quality measurements.

Finally, we focus on the coverage of the Science Europe Evaluation Rubric and show that our additions to the DCS standard increase the coverage of these guidelines and investigate the degree of automation achieved based on the availability of suitable evaluation criteria.

The objective of the evaluation is not to quantitatively evaluate the quality of any given set of maDMPs but to show to what extent the proposed solution can help automate the evaluation of DMPs. Therefore, as long as it is not specified otherwise, when mentioning instances of DMPs we assume the minimal valid DCS DMP shown in Listing 2.1 which is also hosted on Zenodo [1].

[1] https://zenodo.org/records/11670084

6.1 Prototype Evaluation

In this section, we show that the proposed framework meets the requirements outlined in Section 3.5 by providing functional tests for the implemented prototype. To provide a base case for the functional tests, we use the minimal valid DCS DMP shown in the Listing 2.1 as the default input for the test cases, as long as it is not mentioned otherwise. A minimal DMP is sufficient for the evaluation of the prototype as this DMP can be valid and therefore it is an input that enables the solution to address all given use cases.

The tests only address the requirements of the proposed DMP evaluation solution and are independent of the actual resulting quality indicators. These are use-case-specific and therefore consider them in a separate evaluation on the coverage of evaluation goals. It should be noted that these tests are not concerned with correctness of individual components specific to the case study, such as results of an individual evaluation, but rather show that the system is able to provide the functionality demanded by the requirements.

In the remainder of this section, we revisit the given use cases, argue how they have been covered in the proposed conceptual architecture, and verify the efficacy of the prototype by means of functional tests based on the given functional requirements. The given functional tests are system tests that assess the integration of the whole system in a realistic setting. Because we only want to show that the given requirements for the solution are satisfied, one positive test case for each functional requirement is sufficient.

The remainder of the section gives an overview of the test to verify that the prototype addresses all given requirements. Because the outputs of the solution are not human readable we only give an overview of the implemented tests. For details regarding the given tests we refer to the GitHub repository[2] where artifacts of the prototype are hosted, in particular the *evaluation* folder.

Most of the listed tests require the execution of the evaluation process as this integrates the majority of the functionality such as loading of a DMP, preparation of DMP context and the evaluation itself as well as the documentation of relevant evaluation metadata. A full evaluation of a minimal DMP stored on the file-system as *evaluation/minimal.json* for the DMP lifecycle *published* can be triggered using the */api/evaluation/evaluate* Post endpoint of the prototype with the payload listed in Listing 6.1. More documentation regarding possible options is available in the Swagger documentation which is packaged together with the prototype in the corresponding GitHub repository.

[2]https://github.com/larnhold/maDMP-Assesment

```
{
  "dmpLoaderParameters": {
    "dmpLoader": "JSON-FILE",
    "dmpIdentifier": "evaluation/minimal.json"
  },
  "dataLifecycle": "PUBLISHED"
}
```

Listing 6.1: Parameters for the */api/evaluation/evaluate* endpoint of the prototype to trigger the evaluation of the DMP stored on the file system as *evaluation/minimal.json* using all available *Evaluator* components for the DMP lifecycle *PUBLISHED*.

6.1.1 UC1 Generate Measurements

The framework provides a pipeline to provide the necessary information for measurement generation and outsources the evaluation itself to the *Evaluator* components. The requirements associate the functional requirements *FR1-FR6* with this use case, which are listed in Table 6.1, 6.2, 6.3 and 6.4.

Test Name	Load JSON DMP
Description	Load the JSON representation of the minimal DMP using the *JSON DMP Loader* component by invoking an evaluation request. As part of the evaluation process the framework should load the DMP and persist it using the *Data Store* component which writes is to an RDF file.
Expected Result	The resulting RDF file persisted in the *data* folder represents the the given minimal DMP. For reference a corresponding version of the minimal DMP in *Turtle* notation is in the *data/evaluation* sub-folder of the prototype repository. The existence and equivalence of the persisted normalized DMP shows, that the prototype satisfies this requirement.
Achieved Result	The *data* folder contains a *.ttl* file which is equivalent to the reference minimal DMP in Turtle notation.
Test Passed	Yes

Table 6.1: System test for FR1 Load maDMP.

```
{
  "dmpLocations" : [{
    "identifier" : "https://w3id.org/dcso/ns/core#dmp_0",
    "entity" : "https://w3id.org/dcso/ns/core#dmp_0_dataset_0"
  }],
  "sourceIdentifier" : "OPEN_AIRE",
  "value" : '{
    ...
    "description": "The results of the experiment Analysis of
    Performances of kNN and Random Forest. This files contain
    the accuracy, precision, recall and fbeta scores for
    various settings for K-Nearest Neighbors and Random Forest
    for the datasets Zoo (https://www.openml.org/d/62) and
    PhishingWebsites (https://www.openml.org/d/4534).",
    "title": "Results: Analysis of Performances of kNN
    and Random Forest",
    ...
  }',
  "vocabularyIdentifier" : "DATASET"
}
```

Listing 6.2: Excerpt of context fetched from OpenAire containing information regarding the dataset with DOI `10.5281/zenodo.4699026`.

Test Name	Load Dataset Context from OpenAire SKG
Description	When initiating a full evaluation of the minimal DMP using the prototype implementation the *OpenAire Context Loader* component should identify the existence of the included reference to a dataset with DOI `10.5281/zenodo.4699026` and retrieve the corresponding context. As part of the evaluation process this context is then persisted through the *Data Store* component.
Expected Result	The resulting stored DMP context files persisted by the *Data Store* in the *data* folder include the entry retrieved from OpenAire for the given PID. Besides the existence of the required context information, the context should also point to the given dataset with PID `10.5281/zenodo.4699026` in the DMP and specify OpenAire as the source.
Achieved Result	The *data* folder contains a *json* file of the retrieved context. An excerpt of this DMP context is listed in Listing 6.2. This context includes the correct reference to the corresponding DMP and it indicates OpenAire as the source of the information.
Test Passed	Yes

Table 6.2: System test for FR2 Fetch Context and FR3 Connect to SKGs and Repositories and *FR4* Export Context.

```
{
  "lifeCycleStage": {
    "title": "PUBLISHED"
  },
  "isMeasurementOf": {
    "identifier": "required_entity_or_property_existent",
    "description": "Existence of a required entity or property
        according to the specification",
    "title": "DCS Completeness",
    "inDimension": {
      "inCategory": {
        "title": "COMPLETENESS"
      },
      "title": "DCS_COMPLETENESS"
    },
    "applicableDMPLifeCycles": [
      {
        "title": "PLANNING"
      }
    ],
    "expectedDataType": "http://www.w3.org/2001
                        /XMLSchema#boolean",
    "metricTests": []
  },
  "computedOn": {
    "entity": "https://w3id.org/dcso/ns/core#dmp_0"
  },
  "value": True,
  "softwareAgent": {
    "title": "Apache Jena SHACL Validator"
  },
  "testResults": []
}
```

Listing 6.3: DMPQV measurement asserting the completeness of the minimal DMP with regard to the DCS application profile .

Test Name	Produce DCS Completeness Measurements
Description	This test verifies that the solution enables the integration of quality measurements by verifying the case of DCS completeness evaluation given a minimal DMP. The user invokes the evaluation of the minimal DMP with all available *Evaluator* components. After the evaluation the solution returns the result in the form of DMPQV measurements.
Expected Result	Because the minimal DMP contains all entities and fields required by the DCS application profile, the evaluation of DCS completeness should consist of one measurement indicating that the minimal DMP satisfies DCS completeness. As the framework stores the evaluation result in the file system through the *Data Store* component in the folder *data/store*, the persisted file can be accessed to verify the evaluation result. The test should include the assertion of completeness of the given minimal DMP with regard to the DCS application profile which the test verifies by checking if the evaluation result contains a positive DCS completeness measurement.
Achieved Result	The returned measurements contain the DMPQV measurement listed in Listing 6.3, which is a positive DCS completeness measurement.
Test Passed	Yes

Table 6.3: System test for FR5 Produce Measurements.

```
{
  "lifeCycleStage": {
    "title": "PUBLISHED"
  },
  "isMeasurementOf": {
    "identifier": "FsF-F1-01D",
    "description": "Data is assigned a globally
                    unique identifier.",
    "title": "FsF-F1-01D",
    "inDimension": {
      "inCategory": {
        "title": "QUALITY_OF_ACTIONS"
      },
      "title": "FINDABLE",
      "hasParentDimension": {
        "inCategory": {
          "title": "QUALITY_OF_ACTIONS"
        },
        "title": "FAIR"
      }
    },
    "applicableDMPLifeCycles": [
      {
        "title": "PUBLISHED"
      }
    ],
    "expectedDataType": "http://www.w3.org/2001
                         /XMLSchema#integer",
    "metricTests": [
      ...
    ],
    "valueUpperBound": "1",
    "metricGroup": {
      "identifier": "fuji_metric_group",
      "title": "FUJI Metrics",
      "description": ""
    }
  },
  "computedOn": {
    "entity": "https://w3id.org/dcso/ns/core#dmp_0_dataset_0"
  },
  "value": 1,
  "softwareAgent": {
    "title": "FUJI",
    "description": ...
  },
  "testResults": [
    ...
  ]
}
```


Listing 6.4: Result of the F-UJI evaluator for the F-UJI metric-id *FsF-F1-01D* on a dataset with DOI `10.5281/zenodo.4699026` converted to a DMPQV measurement.

Test Name	F-UJI Dataset FAIR Evaluation
Description	The prototype contains the *Achieved FAIRness Dataset Evaluator* which provides FAIR evaluation results for the datasets referenced in the DMP by accessing the external F-UJI FAIR evaluator. The minimal DCS DMP contains the reference to one dataset with the DOI `10.5281/zenodo.4699026`. When the evaluation process is invoked, the prototype orchestrates the execution of the available evaluators, including the *Achieved FAIRness Dataset Evaluator*, and stores the result through the *Data Store* component.
Expected Result	The evaluation result should include the evaluation results from *F-UJI* for the given dataset, converted into the common DMPQV format for quality measurement representation. As with the other functional tests we are not concerned about the correctness of the result as it dependent on on the specific implementation for the given evaluation scenario and in case of the measurements in question even generated by an external service. This test verifies that the resulting 16 measurements are converted from the F-UJI representation to DMPQV and integrated into the overall evaluation result.
Achieved Result	Listing 6.4 shows a resulting measurement which represents the result from the F-UJI evaluator regarding their metric-id *FsF-F1-01D* of the *Findable* FAIR principle. In total 16 measurements with *F-UJI* as the source are included in the list of returned measurements.
Test Passed	Yes

Table 6.4: System test for FR6 Connect to External Evaluators.

6.1.2 UC2 Retrieve Measurements

The proposed framework includes the DMPQV format for the presentation of DMP quality metrics and the corresponding metadata. The test listed in Table 6.5 provides a test that covers the corresponding functional requirements *FR7* and *FR8*. According to these requirements, the solution should be able to provide access to both measurements and corresponding metadata of an evaluation without the need to re-run the evaluation itself.

Test Name	Access Evaluation Result
Description	After running the evaluation with default parameters for the minimal DMP, the prototype returns the the internal identifier of the evaluation in addition to the evaluation measurements in DMPQV form. This identifier references the location where the *Data Store* component persisted the evaluation results. Using this identifier the test queries the corresponding */api/evaluation/{evaluationId}/measurements* endpoint of the prototype to retrieve the stored DMPQV measurements.
Expected Result	The measurements retrieved by calling the corresponding endpoint are equivalent to the ones stored on the filesystem through the *Data Store* and in the DMPQV format. In the prototype implementation the filename is equivalent to the identifier of the evaluation and the storage location is the folder *data/store*.
Achieved Result	The measurements returned by the API are equivalent to the ones stored in the *data/store* folder of the application.
Test Passed	Yes

Table 6.5: System test for FR7 Export Measurements and FR 8 Export Evaluation Metadata.

6.1.3 UC3 Generate Evaluation Report

While the DMPQV quality measurements contain all the information resulting from the DMP evaluation, they might not provide immediate benefit to the reviewer. For example, if a reviewer wants to know how many points a data set scores in a FAIR evaluation or what the mean FAIR score is, then this information needs to be extracted from the available measurements. As a proof of concept, we implemented the calculation of sums and mean values over DMPQV dimensions to show how the information can be processed. Therefore, Table 6.6 lists a test scenario to retrieve the sum and mean value of the measurements in the *availability* dimension and verifies that the values are correct.

This implementation only provides limited possibilities to further process the resulting DMPQV metrics to serve as a proof of concept. To satisfy a wider range of requirements, further work could determine how these measurements could be generated using SPARQL queries.

```json
{
    "evaluationId": "2e24e85c-6174-45f4-ac32-821e7d16ff61",
    "aggregateDimensions": [
        "availability"
    ],
    "averageDimensions": [
        "availability"
    ]
}
```


Listing 6.5: Payload for a POST request to the */api/evaluation/createReport* endpoint to create an evaluation report for the evaluation with id *2e24e85c-6174-45f4-ac32-821e7d16ff61*. The evaluation report should contain the average and the sum of the measurements of the *Availability* dimension.


```json
{
  "dmp": ...,
  "dmpFormat": "RDF/JSON",
  "measurements": ...,
  "sums": {
    "availability": 3.0
  },
  "averages": {
    "availability": 0.5
  }
}
```


Listing 6.6: Result of the evaluation report for an evaluation of the DMP *dcs-repo-examples/ex7-dataset-many.json* from the repository of the prototype containing the sum and the average of the measurement values of the *Availability* dimension.

Test Name	Evaluation Report Including a Sum and an Average
Description	For this test case the evaluation of the DMP *ex7-dataset-many.json* is required as a precondition. This DMP contains 3 references that can be resolved and 3 references which are broken. After initiating the evaluation of this DMP, the reference to the result is returned with which the evaluation report can be requested with the intent to contain the sum and the average of the values of the measurements of the dimensions *Availability*. To get this report the test calls the endpoint */api/evaluation/createReport* of the prototype implementation with the payload as listed in Listing 6.5.
Expected Result	The report contains, besides the DMP and the the harvested context, the requested sum of the values in the *availability* dimensions and the average of these values. Because boolean values are assigned to $false \rightarrow 0$ and $true \rightarrow 1$, the sum is 3 and the average is 0.5, which can be interpreted that 50% of the references are resolvable.
Achieved Result	The resulting report contains the expected DMP and associated context as well as the correct measurements for the sum and average of values in the *Availability* dimension. An excerpt of the result as returned by the prototype is listed in Listing 6.6.
Test Passed	Yes

Table 6.6: System test for FR9 Generate Evaluation Report, FR10 Aggregate Measurements and FR11 Average Measurements.

6.2 Coverage of Evaluation Goals

In this section we discuss to what extent the evaluation goals G1 Completeness, G2 Feasibility, G3 Quality of Actions and G4 Guideline Compliance have been covered in the implemented prototype and what technologies are involved in the implementation to highlight common methods for the implementation of evaluator components. Table 6.7 lists these four evaluation goals and the corresponding evaluation dimensions that address them in the proposed solution.

The goal *G1 Completeness* is distinguished by the two dimensions *DCS Completeness* and *Extension Completeness*. In the prototype implementation, we addressed the *DCS Completeness* with the *DCS Completeness Evaluator* and *Extension Completeness* with the *Science Europe Completeness Evaluator*. Both evaluators provide the required measurements by applying SHACL shapes and are therefore RDF based.

Goal	Dimension	Implemented	Automated	RDF Based	Identifier Analysis	Linked Resources	SKG	FAIR Tools
G1 Completeness	DCS Completeness	✓	✓	✓				
	Extension Completeness	✓	✓	✓				
G2 Feasibility	Accuracy	✓	✓			✓	✓	
	Availability	✓	✓		✓			
	Consistency							
G3 Quality of Actions	FAIR	✓	✓					✓
G4 Compliance	DCS Compliance	✓	✓	✓				
	Extension Compliance	✓	✓	✓				

Table 6.7: Summary of implementation details of evaluation goals and dimensions.

In this thesis, we map the goal *G2 Feasibility* to three dimensions *Accuracy*, *Availability* and *Consistency*. The *Accuracy* dimension is addressed by the *Accuracy Evaluator* which compares the information contained in a DMP with references sources from linked resources such as the Re3Data repository and SKGs such as OpenAire. The *Availability* dimension is covered by the *Availability Evaluator* which analyzed the identifiers provided in a DMP. The *Consistency* dimension has not been addressed because we could not find a sufficient set of rules from the available standards and guidelines, but we also assume that a given DMP is consistent if it follows the DCS standard and also assume that funder extensions do not add inconsistencies.

We set the goal *G3 Quality of Actions* to be equal the FAIR guiding principles and the implemented *Achieved FAIRness Dataset Evaluator* automatically provides measurements of these principles for the datasets referenced in a DMP by requesting FAIR indicators from an external FAIR evaluator and integrates the received measurements into the evaluation results.

In our solution, we cover the evaluation goal *G4 Compliance* with the two dimensions *DCS Compliance* and *Extension Compliance*. The difference between those dimensions is based on a distinction of the underlying source of compliance guidelines, but implementation-wise all implemented evaluators targeting the *Compliance* goal use SHACL shapes to verify whether the DMP meets the given requirements.

6.3 Examples of Quality Measurements

To show the correctness of the resulting quality measurements, in the remainder of this section, we provide examples by manually evaluating the minimal DMP shown in the Listing 2.1 according to a subset of the goals and dimensions implemented in the prototype and compare the resulting measurements with the result of a manual evaluation. This minimal DMP is available in the Github repository of this thesis as *data/case-study/maDMPs/evaluation/minimal.json*. For some cases we alter this minimal DMP and add or remove certain information in order to enable the creation of different quality measurements.

For each dimension considered, we provide positive cases where the DMP meets the requirements and negative cases where we alter the given DMP to introduce an issue that should be detected and included in the evaluation measurements. Together with the introduction of these test cases we also show excerpts of artifacts produced by the prototype solution including instances of the data structures presented in Chapter 4.3.4 such as DMPQV measurements.

6.3.1 G1 Completeness

For the examples regarding the goal *G1* we consider the evaluation of the dimensions *DCS Completeness*, but not *Extension Completeness* as the the generation of measurements for metrics of both dimensions makes use of equivalent methods and only the definition of the underlying guideline is different. We present two examples of DCS completeness evaluation: One positive case where the DMP fulfills all completeness requirements and one negative case where the DMP is missing a required item to be complete with regard to the DCS application profile.

The minimal DMP, by definition, implements all entities and properties required by the DCS application profile and therefore a manually conducted review would assess that this DMP fulfills all completeness requirements requirements given the DCS application profile.

To verify this condition using the prototype produces produces a measurement of the *required_entity_or_property_existent* metric which indicates that the DMP fulfills this requirement. This is a measurement we already discussed in Section 6.1 as part of the verification of the requirements on the implemented prototype and an excerpt is shown in Listing 6.3.

We introduce an error in the given DMP and then verify that the prototype implementation recognizes this error and provides information to assist the human reviewer. Starting from the minimal DMP we delete the included *contact* entity and all underlying properties such as *mbox* and *name* as well as the included sub-entity *contact_id* with it's properties *identifier* and *type*.

From a manual evaluation point of view there are 6 required elements missing in the DMP where the element *contact* is the only one that is a direct successor of an existing element in the DMP, namely the *dmp* root element.

Listing 6.7 shows the measurement produced by the prototype showing that a *Contact* entity for the *Dmp* entity is missing, indicated by the lack of the corresponding relation *hasContact*. The other 5 missing entities are implicitly included in this measurement although the solution cannot provide explicit measurements for them, because the corresponding SHACL queries would be applied to a *Contact* entity and this entity is not existing.

6.3.2 G2 Feasibility

For the evaluation goal *Feasibility* we provide examples of measurements for the implemented dimensions *Accuracy* and *Availability* and show the correctness of each example by comparing the result of a manual evaluation with the corresponding measurements of the automated indicators produced by the prototype.

The minimal DMP used throughout the evaluation contains one dataset with the DOI PID `10.5281/zenodo.4699026` which refers to a dataset hosted on Zenodo. This dataset contains CSV files resulting from an analysis of Performances of kNN and Random Forest. For the evaluation of the *feasibility* category the content of the dataset is not being considered but the properties of the PID and the associated metadata which can be harvested using this identifier are relevant for the evaluation of *Feasibility*.

Accuracy

The minimal DMP does not include sufficient information to provide an illustrative example of accuracy evaluation. Therefore we manually gathered more metadata for the associated dataset with DOI `10.5281/zenodo.4699026` and included the resulting information in the respective entries for this dataset in the minimal DMP. The updated *Dataset* entity is listed in Listing 6.8.

After running the evaluation we can observe that the prototype stores the gathered context in the *data/store* folder relative to the location of the execution. Table 6.8 shows the information gathered from the OpenAire service for the dataset with DOI `10.5281/zenodo.4699026` and the respective name of the property as received from OpenAire as well as the corresponding entity and property from the DCS application value together with its current value in the extended minimal DMP. Similarly, Table 6.9 shows the information harvested for the Host *Zenodo* from re3data. For some cases their exist no corresponding property in the *Host* entity of the DCS standard directly, but this information is contained in higher level entities. Therefore some information received from re3data has to be matched with information contained in the *Dataset* and *Distribution* entities.

```
{
  "measurements":[{
    "lifeCycleStage":{
       "title":"PUBLISHED"
    },
    "isMeasurementOf":{
       "identifier":"required_entity_or_property_existent",
       "description":"Existence of a required entity or
                     property according to the specification",
       "title":"DCS Completeness",
       "inDimension": {
          "title":"DCS_COMPLETENESS",
          ...
       },
       "applicableDMPLifeCycles":...,
       "expectedDataType":"http://www.w3.org/2001
                         /XMLSchema#boolean",
       "metricTests":[]
    },
    "guidance":{
       "title":"SHACL Report",
       "description":"<https://w3id.org/dcso/ns/core#hasContact>:
                     minCount[1]: Invalid cardinality:
                     expected min 1: Got count = 0"
    },
    "computedOn":{
       "entity":"https://w3id.org/dcso/ns/core#dmp_0"
    },
    "value":false,
    "softwareAgent":{
       "title":"Apache Jena SHACL Validator"
    }
  }]
}
```

Listing 6.7: DMPQV *DCS Completeness* measurement indicating the lack of a Contact for the DMP *dmp_0*. The expected amount of *hasContact* relations is 1 but none has been identified.

```
"dataset": [
  {
    "dataset_id": {
      "identifier": "10.5281/zenodo.4699026",
      "type": "doi"
    },
    "title": "Results: Analysis of Performances of kNN
                and Random Forest",
    "personal_data": "unknown",
    "sensitive_data": "unknown",
    "keyword": ["classification", "sklearn", "scikit-learn"],
    "distribution": [{
      "title": "Results: Analysis of Performances of kNN
                and Random Forest",
      "access_url": "https://zenodo.org/records/4699026",
      "data_access": "open",
      "format": ["csv"],
      "license": [{
        "license_ref": "https://creativecommons.org/
                        licenses/by/4.0/legalcode"
      }],
      "host": {
        "url": "https://zenodo.org/",
        "title": "zenodo",
        "pid_system": ["doi"],
        "support_versioning": "yes"
      }
    }],
    "metadata": [{
      "language": "Dublin Core",
      "metadata_standard_id": {
        "identifier": "https://www.dublincore.org/
        specifications/dublin-core/dcmi-terms/",
        "type": "http"
      }
    }]
  }
]
```

Listing 6.8: Additions of manually researched information to augment the information contained in the the minimal DMP regarding the included *Dataset* entity as a basis for automated accuracy evaluation.

Entity	DCSO Property	DCSO Value	OpenAire Property	OpenAire Value
Dataset	dataset_id	10.5281/zenodo.4699026	PID	10.5281/zenodo.4699026
	description	-	Description	-
	language	-	Language	-
	title	Results: Analysis of Performances of kNN8 and Random Forest	Title	Results: Analysis of Performances of kNN8 and Random Forest
	keyword	classification, sklearn, scikit-learn	Subject	classification, sklearn, scikit-learn
Distribution	data_access	open	Access Rights	open
	access_url	https://zenodo.org/	Publisher	https://zenodo.org/
	format	CSV	Format	-
	title	Results: Analysis of Performances of kNN8 and Random Forest	Title	Results: Analysis of Performances of kNN8 and Random Forest
	byte_size	-	Size	-
License	license_ref	https://creativecommons.org/licenses/by/4.0/legalcode	License	https://creativecommons.org/licenses/by/4.0/legalcode

Table 6.8: Instance of mapping of fields from the OpenAire Data Model to the DCS application profile for the dataset with DOI 10.5281/zenodo.4699026.

Entity	DCSO Property	DCSO Value	Re3Data Property	Re3Data Value
Host	pid_system	doi	pidSystem	doi
	url	`https://zenodo.org/`	repositoryURL	`https://zenodo.org/`
	support_versioning	yes	versioning	yes
	certified_with	-	certificate	-
Dataset	metadata	DCMT	metadataStandard	DCMT
Distribution	data_access	open	dataAccess	open

Table 6.9: Instance of mapping of fields from the re3data Data Model to the DCS application profile for the host with URL `https://zenodo.org/`.

Based on the available harvested context, the prototype implementation provides measurements for the accuracy of selected properties contained in the givenDMP by comparing the values stated in the DMP with the corresponding entries in the context according to the proposed mapping. Listing 6.9 shows a resulting DMPQV measurement indicating that the *title* of the referenced dataset with internal reference `https://w3id.org/dcso/ns/core#dmp_0_dataset_0` is accurate, citing the context retrieved from OpenAire as proof.

These measurements are produced for every match between DMP property and mapped context and in case there is a mismatch between the two sources then the corresponding measurement indicates this by setting $value = false$. Additionally in such cases the prototype provides guidance for the reviewer which includes more information on the discrepancy and the expected value. As an example we change the value of the *title* entity of the only dataset contained in the previously mentioned extended minimal DMP with some arbitrary value. After rerunning the evaluation the resulting *accuracy* measurement is similar to the previous measurements of a correct DMP listed in Listing 6.9, but contains the value *false*. Furthermore the measurement contains a *Guidance* entity as listed in 6.10 which explains the negative result of the accuracy measurement for a human reviewer.

```
{
    ...,
    "isMeasurementOf":{
        "identifier":"property_matches_ground_truth_metric",
        "description":"Indicate that the value of a property is
  correct if there is some reference data that can be assumed to
  be true: Compared with OPEN_AIRE",
        "title":"Property matches Ground Truth",
        "inDimension":{
            "inCategory":{
                "title":"FEASIBILITY"
            },
            "title":"ACCURACY"
        },
        ...
        "expectedDataType":"http://www.w3.org/2001/
                                XMLSchema#boolean",
        "metricTests":[]
    },
    "computedOn":{
        "entity":"https://w3id.org/dcso/ns/core#dmp_0_dataset_0",
        "property":"http://purl.org/dc/terms/title"
    },
    "value":"true",
    ...
}
```

Listing 6.9: Measurement showing a successful accuracy measurement of the title of a Dataset by comparing it with a value fetched through the OpenAire context.

Availability

The prototype implementation automatically verifies the availability of all identifiers and URLs mentioned in the DMP to be evaluated by pinging the associated resource and examining the resulting HTTP response code. As an example for measurement of this dimension we discuss measurements targeting the existence of the resource referenced by the identifier for the dataset included in the minimal DMP listed in Listing 2.1. This identifier contains the reference `10.5281/zenodo.4699026` which is a DOI that properly resolves to `https://zenodo.org/records/4699026`. A manual ping of this URL return the HTTP status code *200 OK*. After executing the automated evaluation of the minimal DMP using the prototype implementation, the result contains an accuracy measurement for the mentioned identifier as listed in Listing 6.11. The measurement indicates that the evaluator was successful in verifying the existence of the resource referenced by this identifier.

```
{
    ...,
    "guidance":[
        {
            "title":"DMP value inconsistent with
                    available context.",
            "description":"Value received from OpenAire
                          is 'Results: Analysis of Performances
                          of kNN and Random Forest.'"
        }
    ],
    ...
}
```


Listing 6.10: Guidance included in a negative accuracy measurement giving more information on the encountered discrepancy between DMP value and reference value of a dataset's title.

After changing the *identifier* of this dataset to an arbitrary value, which does not resolve using a DOI resolution service, the resulting measurement indicates this by providing the value *false* for the corresponding *accuracy* measurement for the identifier.

6.3.3 G3 Quality of Actions

The prototype evaluates the goal *Quality of Actions* using the dimensions of the Findable, Accessible, Interoperable, Reusable (FAIR) principles. Furthermore the actual evaluation of these metrics is outsourced to F-UJI, an external service and FAIR evaluator. The prototype assumes that the returned measurements are correct as re-verification of the outputs of F-UJI are out of the scope of this thesis. Therefore we do not provide further examples of FAIR measurements but refer to the work of Devaraju et al. [DH21] for more information on the processing of respective FAIR metrics. In Section 6.1 we show how the prototype of the DMP Evaluator accesses the resulting indicators returned by F-UJI and Listing 6.4 shows an example how these indicators are integrated into the result by converting them into DMPQV measurements.

6.3.4 G4 Guideline Compliance

In the conceptual design of a DMP evaluation solution we proposed 3 dimensions covering the goal G2 Compliance, namely *Guideline Compliance*, *DCS Compliance* and *Extension Compliance*. Because *DCS Compliance* and *Guideline Compliance* share the same types of metrics with the difference only lying in the underlying guidelines used as a reference for the definition of the evaluation metrics. Because the DCS application profile is a well defined standard we therefore give an example of a compliance measurement for the

```
{
    "lifeCycleStage":{
        "title":"PUBLISHED"
    },
    "isMeasurementOf":{
        "identifier":"existence_of_linked_resource_metric",
        "description":"Availability of a linked resource:
                      10.5281/zenodo.4699026",
        "title":"Existence of linked resource",
        "inDimension":{
            "inCategory":{
                "title":"FEASIBILITY"
            },
            "title":"AVAILABILITY"
        },
        "applicableDMPLifeCycles": ...,
        "expectedDataType":"http://www.w3.org/2001
                          /XMLSchema#boolean",
        "metricTests":[]
    },
    "computedOn":{
        "entity":"https://w3id.org/dcso/ns/core
                  #dmp_0_dataset_0",
        "property":"https://w3id.org/dcso/ns/core
                  #dmp_0_dataset_0_datasetId_0"
    },
    "value":"true",
    "softwareAgent": ...
}
```

Listing 6.11: DMPQV measurement indicating the availability of the dataset with DOI `10.5281/zenodo.4699026`.

dimension *DCS Compliance* and not *Extension Compliance*. Furthermore we provide an example for the evaluation dimension *Guideline Compliance* by providing a measurement for a guidance item of the Science Europe DMP Evaluation Rubric.

DCS Compliance

As outlined in the conceptual design in Chapter 4.3.4 we propose to measure *DCS Compliance* using a variety of metrics such as *Multiplicity Constraint*, *Blacklist Vocabulary Constraint*, *Value Pattern Constraint* and *Guideline Verification* as a generic metric. As we showed in Section 6.2 in the reference implementation these metrics are implemented using SHACL queries.

As an illustrative example of *DCS Compliance* evaluation we consider the verification of the correct format of date values in a DCS DMP as an instance of a *Value Pattern Constraint*. The DCS application profile requires the dates to be compliant with *ISO 8601*. As such the value for the field *dmp/created* in the minimal DMP is *2023-10-23T10:10:23.6*, which is an *ISO 8601* compliant string. As with other SHACL based measurement generated by the solution, one DMPQV measurement is created if the given pattern constraint is valid for the entire DMP which results in the measurement listed in Listing 6.12 for the correct minimal DMP.

For the negative test case we change the value of the field dmp/created in the minimal DMP to the value *incorrect*, which is undoubtedly not an ISO8601 compliant date string. As a result the measurements of this altered minimal DMP do no longer state that all date fields are ISO8601 compliant, but instead contains the measurement shown in Listing 6.13, which highlights, that the property *created* of the entity *dmp_0* contains a value which is not DCS compliant and also includes two items of guidance for the reviewer.

Science Europe Guideline Compliance

In the prototype implementation we use the Science Europe DMP evaluation rubric contained in the Practical Guide to the International Alignment of Research Data Management [Eur21] as an instance of funder's DMP requirements. This evaluation rubric contains a multitude of guidelines, out of which only a limited amount contains sufficient evaluation criteria to facilitate an automated evaluation.

As an illustrative example we evaluate the extended minimal DMP, which includes information regarding the distribution and host of the contained dataset, which is available in the repository of the prototype repository as *minimal-with-host.json*. We review a guideline from category 3a which has following description:

> *How will data and metadata be stored and backed up during the research?*

The category contains following guideline which we address in this example:

> *Give preference to the use of robust, managed storage with automatic backup, such as provided by IT support services of the home institution.*

As we outlined in Section 5.3.2, the implementation of the *Evaluator* component for this makes use of the data provided by FAIRSharing[3], as a reference to judge if the use of certain standards or services mentioned in the DMP endorsed by the wider community. Without such a reference it is impossible to answer the given guideline.

According to the proposed mapping of Science Europe guidelines to respective entries in the DCS application profile and the proposed Science Europe extension presented

[3] https://fairsharing.org/

```
{
    ...,
    "isMeasurementOf":{
      "identifier":"dcs_value_pattern_compliance_metric",
      "description":"Restrictions on pattern of values
      according to the DCS application profile",
      "title":"DCS Value Pattern Constraint",
      "inDimension":{
        "inCategory":{
          "title":"COMPLIANCE"
        },
        "title":"DCS_COMPLIANCE",
        "hasParentDimension":{
          "inCategory":{
            "title":"COMPLIANCE"
          },
          "title":"GUIDELINE_COMPLIANCE"
        }
      },
      ...,
      "expectedDataType":"http://www.w3.org/2001/
      XMLSchema#boolean",
      "metricTests":[]
    },
    "computedOn":{
      "entity":"https://w3id.org/dcso/ns/core#dmp_0",
      "property":""
    },
    "value":True,
    "softwareAgent":{
      "title":"Apache Jena SHACL Validator"
    },
    "testResults":[]
}
```

Listing 6.12: DMPQV measurement indicating that all fields in the DMP that should be in ISO8601 format according to the DCS standard are valid.

```
{
  ...,
  "isMeasurementOf":{
    "identifier":"dcs_value_pattern_compliance_metric",
    ...
    "inDimension":{
      "inCategory":{
        "title":"COMPLIANCE"
      },
      "title":"DCS_COMPLIANCE",
      ...
    },
    "applicableDMPLifeCycles": ...,
    "expectedDataType":"http://www.w3.org/2001/
                        XMLSchema#boolean",
    "metricTests":[]
  },
  "guidance":[
    {
      "title":"SHACL Report",
      "description":"<https://w3id.org/dcso/ns/core#created>:
      Pattern[^[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}
      :[0-9]{2}(Z|[+-][0-9]{2}:[0-9]{2})$]:
      Does not match: 'created'"
    },
    {
      "title":"Invalid Date",
      "description":"The given value is not
      an ISO8601 compliant date"
    }
  ],
  "computedOn":{
    "entity":"https://w3id.org/dcso/ns/core#dmp_0",
    "property":"https://w3id.org/dcso/ns/core#created"
  },
  "value":False,
  "softwareAgent":{
    "title":"Apache Jena SHACL Validator"
  }
}
```

Listing 6.13: DMPQV measurement highlighting that the value for the property *created* of the entity *dmp_0* is not an ISO8601 compliant date.

in Section 5.3.2, the information necessary to evaluate this question is present in the *distribution/host* property defined by the DCS. The corresponding reference values of community accredited hosts are represented by the FAIRSharing entity *Database*. The host value given in the DMP to be evaluated is *zenodo* and a manual lookup confirms that this host is registered in FAIRSharing with the DOI URL `https://doi.org/10.25504/FAIRsharing.wy4egf`. Therefore the result of a manual evaluation is, that the given DMP fulfills the given guideline of giving preference to robust storage.

Similarly the evaluation result from the automated evaluation of the prototype contains the measurement shown in Listing 6.14 which indicates the same result thorough the value *true*. What is noticeable about this particular measurement is the use of nested *Evaluation Dimension* entities in order to properly model the hierarchy of the of the given evaluation rubric. Through the use of sub-dimension, all structural information from the requirement of the guideline is persisted in the measurement while the metric itself is based on the metric with identifier *science_europe_guideline_verification_metric* which can help machines interpret the measurement.

If we change the name of the *host* in the DMP to a value which has no corresponding entry in the FAIRSharing registry, then the resulting measurement correspondingly indicates this by having a a value of *false*.

6.4 Evaluation of Science Europe DMP Evaluation Rubric

In this section, we evaluate the degree to which the proposed Science Europe funder extension increases the coverage of guidelines expressed by the corresponding Science Europe DMP Evaluation Rubric [Eur21]. In addition to showing the utility of the proposed funder extension, we also discuss the viability of these guidelines as a baseline for automated indicators.

To this extent, we evaluate to what extent the elements in the Science Europe Evaluation Rubric can be represented by a mapping to elements in the DCS application profile and then compare this result with a mapping of the same guideline items to the union of DCS application profile and the proposed Science Europe funder extension.

We define that a guideline is largely covered if the information required by a guideline can clearly be represented by the selected entry in the DMP. If the mapping is not obvious but it can still be argued that the information required by the guideline could be contained in a field in the DMP then we mark this mapping as partially covered. In all other cases, the selected guideline is marked as not covered.

The proposed alignment of elements of the Science Europe Evaluation Rubric to the DCS application profile and the Science Europe Funder Extension is independent of the semantics of the guideline and independent of the ability of the mapping to automatically provide an answer based on the semantic meaning of the guideline in question. Therefore, we provide a strict distinction between the possibility of mapping and the possibility of automated evaluation.

```
{
    ...,
    "isMeasurementOf":{
        "identifier":"science_europe_guideline
        _verification_metric",
        "description":"Indicates if a guideline of the S
        cience Europe DMP Evaluation Rubric
        is fulfilled or not.",
        "title":"Science Europe Guideline Verification",
        "inDimension":{
            "title":"Give preference to the use of robust,
            managed storage with automatic backup, such as
            provided by IT support services of the
            home institution.",
            "hasParentDimension":{
                "title":"How will data and metadata be stored
                and backed up during the research?",
                "hasParentDimension":{
                    "title":"SCIENCE_EUROPE_GUIDELINE
                                    _COMPLIANCE",
                    "hasParentDimension":{
                        "inCategory":{
                            "title":"COMPLIANCE"
                        },
                        "title":"GUIDELINE_COMPLIANCE"
                    }
                }
            }
        },
        "applicableDMPLifeCycles":[
            {
                "title":"PLANNING"
            }
        ],
        "expectedDataType":"http://www.w3.org/2001/
                            XMLSchema#boolean",
        "metricTests":[]
    },
    "computedOn":{
        "entity":"https://w3id.org/dcso/ns/core#
                dmp_0_dataset_0_distribution_0_host_0"
    },
    "value":"true",
    ...
}
```

Listing 6.14: Resulting DMPQV measurement of the automated evaluation of a guideline of the Science Europe DMP evaluation rubric. The hierarchy of the entries in the evaluation rubric is modelled through the use of nested *Evaluation Dimension* entities

Category	Largely Covered		Partially Covered		Not Covered		#
General Information	1	100,00%	0	0,00%	0	0,00%	1
Data Description and Collection or Reuse of Existing Data	4	44,44%	2	22,22%	3	33,33%	9
Documentation and Data Quality	6	85,71%	0	0,00%	1	14,29%	7
Storage and Backup During the Research Process	4	66,67%	0	0,00%	2	33,33%	6
Legal and Ethical Requirements, Codes of Conduct	3	50,00%	2	33,33%	1	16,67%	6
Data Sharing and Long-Term Preservation	8	66,67%	4	33,33%	0	0,00%	12
Data Management Responsibilities and Resources	6	85,71%	1	14,29%	0	0,00%	7
Total	**32**	**66,67%**	**9**	**18,75%**	**7**	**14,58%**	**48**

Table 6.10: Extent coverage of the Science Europe DMP Evaluation Rubric by the DCS Standard.

The result of our mapping is shown in Table 6.10. In addition to the $66,67\%$ that can be unambiguously assigned to an entry in the DCS standard, there are also $18,75$ that do not have a distinct correlation with any DCS field, but the information can be assumed to be contained in a more generic field and could therefore be partially covered. For $14,58\%$ of the items in the Evaluation Rubric we were unable to deduce a mapping to the DCS profile.

To increase the coverage of the DCS application profile, Cardoso et al. [CJM+20] proposed an extension for the DCS standard to increase coverage of a variety of funder templates. As a consequence of the implementation of a prototype of the DMP evaluation framework proposed in this thesis, we extended this proposed funder extension in Section 5.1.1 to specifically provide better coverage of the Science Europe DMP evaluation rubric. Table 6.11 shows the extent to which the union of DCS profile and the extension cover the items of the Science Europe Evaluation Rubric based on our assesment. By adding properties to the extension that explicitly address items in the Evaluation Rubric, all items can be mapped to a corresponding field in the DMP. Despite this perfect coverage, it should be noted that this value represents just the coverage and gives no indication of the quality of the evaluation that this mapping enables.

Category	Largely Covered		Partially Covered		Not Covered		#
General Information	1	100,00%	0	0,00%	0	0,00%	1
Data Description and Collection or Reuse of Existing Data	9	100,00%	0	0,00%	0	0,00%	9
Documentation and Data Quality	7	100,00%	0	0,00%	0	0,00%	7
Storage and Backup During the Research Process	6	100,00%	0	0,00%	0	0,00%	6
Legal and Ethical Requirements, Codes of Conduct	6	100,00%	0	0,00%	0	00,00%	6
Data Sharing and Long-Term Preservation	12	100,00%	0	0,00%	0	0,00%	12
Data Management Responsibilities and Resources	7	100,00%	0	0,00%	0	0,00%	7
Total	**48**	**100,00%**	**0**	**0,00%**	**0**	**0,00%**	**48**

Table 6.11: Extent coverage of the Science Europe DMP Evaluation Rubric by adding the proposed Science Europe Funder extension to a DCS DMP.

Lastly, we discuss the viability of the Science Europe DMP Evaluation Rubric as a baseline for the definition of DMPQV metrics. In Section 5.3.2 we implemented an *Evaluator* component with the intention of automatically providing indicators that address the questions of the rubric. As part of the creation of these indicators, an evaluation condition is required as an interpretation when a certain guideline is fulfilled by the DMP. However, as Table 6.12 highlights, only a fraction of the items in the guideline provide a condition that is sufficient to provide an interpreted result. This is the case because most of the guidelines only ask for the existence of some information, which can be verified by checking the existence of the corresponding mapped field in the DMP. Only 4 out of 48 distinct items in the guideline specify requirements in addition to the mere existence of some information.

Therefore, the Science Europe DMP Evaluation Rubric has only limited use as a baseline for automated indicators due to the lack of machine-actionable conditions that specify when a DMP fulfills a guideline. Future work can focus on specifying conditions that can automatically be evaluated to truly enable automated evaluation of funder guidelines.

Category	**Sufficient Evaluation Condition**		**#**
General Information	0	0,00%	1
Data Description and Collection or Reuse of Existing Data	1	11,11%	9
Documentation and Data Quality	1	14,29%	7
Storage and Backup During the Research Process	1	16,67%	6
Legal and Ethical Requirements, Codes of Conduct	0	0,00%	6
Data Sharing and Long-Term Preservation	1	8,33%	12
Data Management Responsibilities and Resources	0	0,00%	7
Total	**4**	**8,33%**	**48**

Table 6.12: Extent of items in the Science Europe Evaluation Rubric with sufficient evaluation conditions to provide automated indicators besides verification of existence of a property.

CHAPTER 7

Conclusion and Future Work

DMPs aim to promote good data management practices in scientific and research contexts. Initially being a static text document required by funders when applying for grants, the focus of DMPs has shifted to machine actionability in order to make DMPs a living document that can add value in different phases of the research data life cycle. Miksa et.al [MWN+21] proposed the DCS application profile which is the RDA standard for maDMPs. As a result, various aspects of RDM could already be automated, reducing administrative burden and improving processes involved in RDM. Still, to our knowledge there exists no automated solution that provides the ability to evaluate the contents in a maDMP including the requirements given by funding bodies.

In this thesis, we explored the possibilities of automated evaluation of DMPs both from a general perspective and from the perspective of compliance with funder requirements. After investigating the literature in the relevant fields of data quality standards, FAIR evaluation, semantic web technologies and DMPs, we collected requirements for a solution for the automated evaluation of DMPs, based on previous work on maDMPs and community requirements to address the problem stated in the motivation of this thesis. We proposed a conceptual design for a framework for the automated evaluation of DMPs that addresses the given requirements as well as a taxonomy of evaluation goals and dimensions and corresponding types of evaluation metrics. Additional artifacts include a proposal of the DMPQV vocabulary for the communication of DMP quality measurements as well as a mechanism to represent contextual information for a given maDMP.

To evaluate the proposed framework, we implemented a prototype of the solution as a Java Spring application and provided a case study for the evaluation of the guidelines published by Science Europe in the Practical Guide to the International Alignment of Research Data Management. To cover the guidelines contained in this evaluation rubric, we proposed an extension to the DCS standard. In addition to the implementation of Evaluators that address these requirements, we also considered funder independent metrics derived from the evaluation goals of *Completeness*, *Feasibility*, *Quality of Actions* and *Compliance*. Furthermore, we provided examples of DMP evaluation scenarios and compared the resulting DMPQV measurements with the results of a manual assessment.

The result of the evaluation shows that the prototype of the proposed solution meets the requirements and addresses the given use cases and functional requirements. The proposed solution is capable of automatically generating quality measurements and evaluation reports and provides access to the information produced in a standardized format. However, while the solution can automatically produce these indicators, the quality of the measurements is dependent on the available guidelines, and while we are able to cover all information required by the Science Europe Evaluation Rubric with the proposed extension, more precise evaluation criteria are needed to improve the meaningfulness of the evaluation result.

7.0.1 Review of Research Questions

In this section we revisit the research questions defined in the introduction of this thesis. We summarize how they have been answered and provide references to relevant parts of the thesis.

RQ1 To what extent can maDMPs be automatically validated with respect to goals G1 - G4?

In Section 2.2 we investigated what kind of data quality metrics have been proposed in literature and align them with the DMP evaluation goals G1 Completeness, G2 Feasibility, G3 Quality of Actions and G4 Compliance proposed by Miksa et al. [MSS+23]. Based on this alignment, we proposed a taxonomy in Section 4.1.2 that assigns each DMP evaluation goal to an evaluation category and each category the corresponding evaluation dimensions, which is shown in Figure 4.6. For each of these dimensions, we proposed templates of indicators that characterize parts of an evaluation dimension.

As part of the implementation of a prototype, in Section 5.3.2, we provided details on the implementation of the evaluator components responsible for the generation of the proposed DMP quality measurements based on the proposed metric types. The measurement generation processes utilizes different methods and technologies which we summarized in Table 6.7. We therefore showed how semantic web technologies, SKGs and FAIR evaluation tools can be leveraged to provide these measurements. Furthermore, all the measurements provided can be automatically produced, although human interpretation is necessary to provide a final evaluation result.

Furthermore, in Section 6.3, we provide examples of measurements of automated indicators for the given review goals which have been produced by the reference implementation and compare them with the results of corresponding manual assessments.

RQ2 To what extent can the coverage of funder guidelines according to G4 Guideline Compliance be increased by adapting the RDA DMP Common Standard?

We identified the Science Europe DMP Evaluation Rubric as a DMP guideline that represents the requirements of major European funding organizations. Cardoso et al. [CJM+20] proposed an extension to increase the coverage of the most prominent DMP funder templates and in Section 2.5.1 we gave an overview of this proposal as a starting point for further adaptations. Based on this preliminary work, we proposed further additions to this extension in Section 5.1.1 to increase the coverage of the information required by the Science Europe DMP guidelines. In Section 6.4 we evaluated the coverage of the proposed extension by mapping the corresponding question in the Science Europe DMP Evaluation Rubric and show that each given question can be mapped and answered by a corresponding entry in the DCS application profile or the proposed funder extension. Therefore we where able to extend the coverage of the Science Europe DMP evaluation rubric by a maDMP to 100%.

RQ3 What is a conceptual framework that provides automated indicators that measure aspects of the quality of a maDMP at different stages of the research data lifecycle?

A main contribution of this thesis is the conceptual design of a framework for the automated evaluation of DMPs in Chapter 4.3.4 based on the requirements elaborated in Chapter 3.5. We described the reference architecture using parts of the TOGAF EA framework and proposed a business architecture and an information systems architecture based on the arc42 architecture documentation template.

As part of the proposed business architecture, we provided an overview of the processes involved in the automated evaluation of DMPs from a nontechnical point of view. We showed what steps are involved in the automated evaluation of DMPs and proposed a taxonomy of evaluation categories, dimensions, and metric templates as a reference for implementations.

As part of the information systems architecture, we described the data architecture, as well as the application architecture of the conceptual framework. In the scope of the data architecture, we proposed the Data Management Plan Quality Vocabulary (DMPQV) in Section 4.2.1 as a means to represent DMP quality measurements and the corresponding metadata based on a set of competency questions. This vocabulary also considers the generation of metric measurements at different stages of the DMP lifecycle. Furthermore, in Section 4.2.2, we proposed a structure to provide contextual information to augment the content of a DMP and make referenced information explicitly available.

In Section 4.3 we outlined the application architecture for a conceptual framework for the automated evaluation of DMPs which is able to produce DMPQV measurements for the different stages of the DMP lifecycle. Due to the modular architecture it is possible to dynamically invoke evaluators responsible for the given DMP lifecycle. Following the arc42 architecture template, we described the components of the framework from various levels of abstraction, as well as their interactions associated with the fulfillment of the given requirements.

In Chapter 5.4 we showed the feasibility of the proposed framework by implementing a prototype. In addition to instantiating the artifacts proposed in the conceptual architecture, we conducted a case study to provide automated indicators regarding the DMP requirements given by the Science Europe DMP Evaluation Rubric and extended the implementation accordingly.

Finally, in Chapter 6.4, we evaluated the efficacy of the proposed conceptual reference architecture by providing functional tests for the prototype, to validate the coverage of the use cases and functional requirements outlined in Chapter 3.5 and discuss how the DMP evaluation goals have been implemented in the prototype. Furthermore, we evaluated the case study by comparing the coverage and achieved automation of the evaluation of the Science Europe DMP guidelines and compare our results with previous studies.

7.1 Future Work

In this thesis we investigated the possibilities of automated evaluation of DMPs based on previous work of Miska et al. [MSS+23]. Although our proposed work provides a more detailed view of this topic and the proposed solution covers a wide range of evaluation requirements, we also identified further topics that can be explored in this area.

Our solution provides automated measurements in the DMPQV format. But the proposed and implemented metrics share the constraints that only machine-actionable parts are considered to also mitigate the issue of automation bias. As Miksa et al. [MSS+23] point out, the integration of non-machine-actionable parts of a DMP could be included in the evaluation to provide further insight. As such, for example, free-form text could be included in the evaluation using NLP technologies. This raises the question of how free-form text should be verified and if the solution should automatically provide an opinion on the values or just highlight and extract relevant parts for later manual review.

Currently, SHACL shapes verifying the requirements of the DCS application profile and extensions are manually derived from the corresponding standard. The issue with this approach is that if the standard changes, then the shapes need to be adapted, and also the integration of new shapes for new extensions might contain repetitive work that could be automated. Cimmino et al. [CFIGC20] investigate the automated creation of SHACL shapes which could be adapted to be used to automatically create verification shapes for the evaluation of DMPs from the respective standards.

The proposed solution provides the ability to integrate context from external sources into the evaluation process. In the prototype, we showed how context from the OpenAire SKG and the Re3Data repository can be integrated as DMP context. But we also highlighted that these sources provide the information in non-uniform formats and while standards like SKG-IF[1] have already been proposed for SKGs they have yet to be implemented. Another notable approach is the Research Object Crate (RO-Crate)[SRSC+22] which facilitates the packaging of research artifacts annotated with schema.org metadata[2]. This unification of standards could also benefit the proposed solution as it would make it possible to source contextual information from a wider variety of sources and a common format would help align it with the corresponding fields in the DMP. Another type of resource provider that could be considered in future developments is the integration of CRIS systems to retrieve institution-specific context.

In the proposed solution we adapted the W3C Data Quality Vocabulary (DQV) and proposed the Data Management Plan Quality Vocabulary (DMPQV) to store the evaluation results in the form of quality measurements and related metadata. Future work could investigate the inclusion of other relevant vocabularies and how they could be adapted in the context of DMP evaluation. One potentially useful vocabulary is the Open Digital Rights Language (ODRL), a policy expression language to represent statements about content and service usage.

Currently, the use case for report generation only includes summarizing and calculating the averages of the measurement values of a dimension to show how the resulting DMPQV measurements can be interpreted. Future work can extend this functionality and investigate how the measurements can be processed to provide more accessible evaluation results. As the DMPQV results are available in RDF format, the use of SPARQL queries would be suitable. This processing could also include more higher-level abstractions such as DMPQV categories and metric groups.

For the case study, we used Science Europe DMP Evaluation Rubric as a baseline to represent funder requirements and proposed an extension to model these requirements. But with increasing focus on RDM, a variety of other evaluation rubrics have emerged that aim to standardize the evaluation of DMPs which could be considered in future adoptions of the proposed solution. As such, the DART project summarizes the requirements of the United States National Science Foundation after conducting a review of submitted DMPs [Car17, WCW+16]. Another publication summarizing DMP requirements is the Belmont Forum Data Management Plan Scorecard [BUD+19]. Furthermore, this thesis only considered a subset of quality dimensions proposed in the literature that have been identified as important when reviewing DMPs in previous work [MSS+23]. More quality dimensions could be identified and taken into account to provide a wider range of metrics to reviewers.

The proposed solution relies heavily on SHACL to verify if the given DMP complies with the given evaluation requirements. For this thesis, efficiency has not been taken into account as the assumption is that only one DMP will be reviewed at a time. For other applications, larger quantities of DMPs may have to be reviewed, such as when automatically evaluating all DMPs of a repository. In such cases performance might

[1] https://skg-if.readthedocs.io/

[2] https://schema.org/

be an issue and Roblado et al. [RPZ+23] propose a faster solution than SHACL called DLV2 that could improve the efficiency of the solution.

The majority of currently available maDMPs is currently available in JSON notation. In our prototype implementation, we implemented a component to convert DCS DMPs to DCSO as an RDF graph. This conversion also includes the proposed Science Europe Funder extension. However, the provided component is tailored to the given standard and needs to be adapted when the DCS format changes. Future work could explore more flexible approaches of JSON to RDF conversion. As such, the maDMP-Ontology Toolkit (MadPot)[3] provides converison of JSON maDMPs to RDF by exploiting JSON-LD. By appending the corresponding headers to the JSON document it can automatically be converted to RDF. But during our tests, we found out that the current solution does not provide sufficient mapping, as especially the relation between entities is not precisely converted. However, the approach is promising, and future work can improve the already existing solution.

In the prototype implementation, all components are integrated in the scope of a single Spring application. This approach is sufficient to show the feasibility of the proposed architecture, but in an applied scenario due to different evaluation requirements of different communities, different components to load DMPs, harvest evaluation context and provide quality measurements might only be available in a distributed fashion. For example, the FAIR evaluator [WDS+19] provides a flexible way to integrate externally hosted evaluators through the use of SmartAPIs[4]. The prototype implementation of the proposed DMP evaluator provides a flexible architecture that enables future changes to the resolution of the evaluator components so that future work could explore the integration of externally hosted instances of components proposed in the reference architecture.

This thesis mainly explored the possibilities of DMP evaluation from a technical perspective. Future work could apply the proposed solution in an applied context collect feedback for the further development of the solution in order to improve the available metrics to better cover the evaluation requirements. This could include case studies with researchers and funding bodies.

The proposed evaluation solution provides the ability to automate the generation of quality measurements. In some cases automated generation is not possible because the assistance of a human reviewer is needed. This could be the case when funder requirements are phrased in a way that requires rather subjective judgement and no objective guidelines are given that would be sufficient for automated evaluation. Future work could investigate the use of human-in-the-loop approaches and adapt the solution accordingly. This requires changes in the proposed architecture to account for the time a human reviewer needs for the review and the thereby introduced asynchronous computation aspect. Another approach could be to add another independent component to further process the available measurements and include human actions.

[3] `https://github.com/fekaputra/MadPot`
[4] `https://smart-api.info/`

The solution proposed in this thesis can be used for future studies on the quality and compliance aspects of DMPs to monitor the adoption of good RDM practices. To our knowledge, there exists no large dataset of DCS DMPs but different providers like DMPTool[5] and ARGOS[6] provide publicly available plans that could be compiled into a dataset of DCS DMPs to be evaluated with the proposed solution to provide an overview of the current quality of DMPs.

The proposed solution does not exist in a vacuum and there are parallels to other initiatives for the evaluation of digital objects such as FAIR evaluators. Currently, different FAIR evaluators return their results using different formats, and to our knowledge, there exists no uniform standard. With the proposal of DMPQV we proposed a standard for the representation of DMP quality measurements which also covers results of FAIR evaluator as we showed with the inclusion of the results of the F-UJI evaluator. In the future the FAIR community plans to propose their own standard for the representation of FAIR metrics and measurements. In that case, the proposed DMPQV needs to be adapted to comply with the work of the FAIR community.

[5] https://dmptool.org/public_plans
[6] https://argos.openaire.eu/explore-plans

APPENDIX A

Resources

A.1 Prototype Source Code

The artifacts implemented as part of this thesis, including the DMPQV schema, the Science Europe funder extension ontology, as well as the implementation of the DMP Evaluation Service prototype are available in the following GitHub repository:

`https://github.com/larnhold/maDMP-Assesment`

For more information on how to setup and use this implementation we provide extensive documentation in the *README.md* file of this repository.

A.2 DMPQV Documentation

We created a documentation for the RDF implementation of the proposed Data Management Plan Quality Vocabulary (DMPQV) data structure for the representation of quality measurements for DMPs from the conceptual design proposed in Section 4.2.1. The documentation is hosted on `https://w3id.org/dmpqv` and provides besides a extensive explanation of the included elements the possibility to download the RDF ontology in various formats.

A.3 Science Europe Evaluation Rubric Mapping

In order to have a reference in the DMP for all information required to address the guidelines in the Science Europe Evaluation Rubric we proposed an extension for the DCS application profile. The documentation and the underlying ontology are available at `https://w3id.org/dcs-se-extension`.

Table A.1 shows the mapping of guidelines to respective entries in the DCS application profile or the Science Europe funder extension proposed in Section 5.1.1.

Table A.1: Mapping of Science Europe DMP Evaluation Rubric Guidelines to the DCS Application Profile and the Science Europe Extension proposed in this Thesis.

Guideline	**DCS Mapping**	**Science Europe Funder Extension Mapping**
Provide information such as name of applicant, project number, funding programme, version of DMP.Provide information such as name of applicant, project number, funding programme, version of DMP.	dmp/ethical_issues_exist, dmp/ethical_issues_description dmp/ethical_issues_report	
Explain which methodologies or software will be used if new data are collected or produced.		dataset/methodology
State any constraints on re-use of existing data if there are any.	ditribution/data_access dmp/ethical_issues_exist	dataset/is_reused distribution/restriction_explanation
Explain how data provenance will be documented.		dataset/data_quality_assurance
Briefly state the reasons if the re-use of any existing data sources has been considered but discarded.		dataset/methodology
Give details on the kind of data: for example, numeric (databases, spreadsheets), textual (documents), image, audio, video, and/or mixed media.	dataset/type dataset/description	
Give details on the data format: the way in which the data is encoded for storage, often reflected by the filename extension (for example pdf, xls, doc, txt, or rdf).	dataset/format distribution/format	
Justify the use of certain formats. For example, decisions may be based on staff expertise within the host organisation, a preference for open formats, standards accepted by data repositories, widespread usage within the research community, or on the software or equipment that will be used.	distribution/description	distribution/format_justification
Give preference to open and standard formats as they facilitate sharing and long-term re-use of data (several repositories provide lists of such 'preferred formats').	distribution/format	
Give details on the volumes (they can be expressed in storage space required (bytes), and/or in numbers of objects, files, rows, and columns).	distribution/byte_size	
Indicate which metadata will be provided to help others identify and discover the data.	dataset/metadata	
Indicate which metadata standards (for example DDI, TEI, EML, MARC, CMDI) will be used.	dataset/metadata	
Use community metadata standards where these are in place.	dataset/metadata	

Continued on next page

Table A.1: Mapping of Science Europe DMP Evaluation Rubric Guidelines to the DCS Application Profile and the Science Europe Extension proposed in this Thesis. (Continued)

Indicate how the data will be organised during the project mentioning, for example, conventions, version control, and folder structures. Consistent, well-ordered research data will be easier to find, understand, and re-use.	dataset/metadata dataset/keyword dataset/data_quality_assurance	
Consider what other documentation is needed to enable re-use. This may include information on the methodology used to collect the data, analytical and procedural information, definitions of variables, units of measurement, and so on.		dataset/methodology
Consider how this information will be captured and where it will be recorded (for example in a database with links to each item, a 'readme' text file, file headers, code books, or lab notebooks).	metadata/description distribution, host	
Explain how the consistency and quality of data collection will be controlled and documented. This may include processes such as calibration, repeated samples or measurements, standardised data capture, data entry validation, peer review of data, or representation with controlled vocabularies.	dataset/data_quality_assurance	
Describe where the data will be stored and backed up during research activities and how often the backup will be performed. It is recommended to store data in least at two separate locations.	host/backup_type host/backup_frequency	
Give preference to the use of robust, managed storage with automatic backup, such as provided by IT support services of the home institution. Storing data on laptops, stand-alone hard drives, or external storage devices such as USB sticks is not recommended.	host/id	
Explain how the data will be recovered in the event of an incident.		host/description host/data_recovery_explanation
Explain who will have access to the data during the research and how access to data is controlled, especially in collaborative partnerships.	dataset/security_and_privacy	
Consider data protection, particularly if your data is sensitive (for example containing personal data, politically sensitive information, or trade secrets). Describe the main risks and how these will be managed.	dataset/ethical_issues_description dataset/sensistive_data	
Explain which institutional data protection policies are in place.	dataset/security_and_privacy	dmp/related_policies
Ensure that when dealing with personal data, data protection laws (for example GDPR) are complied with. (including sub-points)	dataset/security_and_privacy	distribution/property_rights_explanation

Continued on next page

Table A.1: Mapping of Science Europe DMP Evaluation Rubric Guidelines to the DCS Application Profile and the Science Europe Extension proposed in this Thesis. (Continued)

Explain who will be the owner of the data, meaning who will have the rights to control access. (including sub-points)	contributor/role	
Indicate whether intellectual property rights (for example Database Directive, sui generis rights) are affected. If so, explain which and how will they be dealt with.		distribution/property_rights_explanation
Indicate whether there are any restrictions on the re-use of third-party data.	distribution/data_access distribution/licence	distribution/restriction_explanation
Consider whether ethical issues can affect how data are stored and transferred, who can see or use them, and how long they are kept. Demonstrate awareness of these aspects and respective planning.	dmp/ethical_issues_exist dmp/ethical_issues_description dmp/ethical_issues_report	
Follow the national and international codes of conducts and institutional ethical guidelines, and check if ethical review (for example by an ethics committee) is required for data collection in the research project.	dmp/ethical_issues_exist dmp/ethical_issues_description dmp/ethical_issues_report	
Explain how the data will be discoverable and shared (for example by deposit in a trustworthy data repository, indexed in a catalogue, use of a secure data service, direct handling of data requests, or use of another mechanism).	host/titl host/url distribution/description distribution/data_access	
Outline the plan for data preservation and give information on how long the data will be retained.	distribution/available_until dataset/preservation_statement	
Explain when the data will be made available. Indicate the expected timely release. Explain whether exclusive use of the data will be claimed and if so, why and for how long. Indicate whether data sharing will be postponed or restricted for example to publish, protect intellectual property, or seek patents.	distribution/license license/start_date	
Indicate who will be able to use the data. If it is necessary to restrict access to certain communities or to apply a data sharing agreement, explain how and why. Explain what action will be taken to overcome or to minimise restrictions.	distribution/licence dmp/contributor	distribution/restriction_explanation
Indicate what data must be retained or destroyed for contractual, legal, or regulatory purposes.	dataset/preservation_statement	dmp/related_policies
Indicate how it will be decided what data to keep. Describe the data to be preserved long-term.	dataset/preservation_statement	
Explain the foreseeable research uses (and/or users) for the data.	dataset/description	

Continued on next page

Table A.1: Mapping of Science Europe DMP Evaluation Rubric Guidelines to the DCS Application Profile and the Science Europe Extension proposed in this Thesis. (Continued)

Indicate where the data will be deposited. If no established repository is proposed, demonstrate in the DMP that the data can be curated effectively beyond the lifetime of the grant. It is recommended to demonstrate that the repositories policies and procedures (including any metadata standards, and costs involved) have been checked.	dmp/cost dataset/preservation_statement	
Indicate whether potential users need specific tools to access and (re-)use the data. Consider the sustainability of software needed for accessing the data.	dataset/technical_resource	dataset/methodology
Indicate whether data will be shared via a repository, requests handled directly, or whether another mechanism will be used?	dataset/technical_resource	dataset/methodology
Explain how the data might be re-used in other contexts. Persistent identifiers (PIDs) should be applied so that data can be reliably and efficiently located and referred to. PIDs also help to track citations and re-use.	dataset/dataset_id	
Indicate whether a PID for the data will be pursued. Typically, a trustworthy, long-term repository will provide a persistent identifier.	dataset/dataset_id	
Outline the roles and responsibilities for data management/ stewardship activities for example data capture, metadata production, data quality, storage and backup, data archiving, and data sharing. Name responsible individual(s) where possible.	contributor/role	
For collaborative projects, explain the co-ordination of data management responsibilities across partners	contributor/role	
Indicate who is responsible for implementing the DMP, and for ensuring it is reviewed and, if necessary, revised.	contributor/role	
Consider regular updates of the DMP.	dmp/modified	
Explain how the necessary resources (for example time) to prepare the data for sharing/preservation (data curation) have been costed in.	dmp/cost	dataset/methodology
Carefully consider and justify any resources needed to deliver the data. These may include storage costs, hardware, staff time, costs of preparing data for deposit, and repository charges.	dmp/cost	
Indicate whether additional resources will be needed to prepare data for deposit or to meet any charges from data repositories. If yes, explain how much is needed and how such costs will be covered.	dmp/cost dataset/technical_resource	

List of Figures

List of Tables

List of Listings

Acronyms

API Application Programming Interface. 40, 76, 81–83, 90, 91, 94, 95, 105

BPMN Business Process Model and Notation. 48

CRIS Current Research Information System. 131

CSV Comma-separated values. 110, 113

DC Dublin Core. 57, 61

DCAT Data Catalog Vocabulary. 83

DCC Digital Curation Centre. 25

DCMT Dublin Core Metadata Terms. 114

DCS DMP Common Standard. ix, xi, 1, 3, 5, 6, 8, 27–31, 33, 42, 49, 51–55, 59, 62, 64, 73, 74, 76, 78–81, 83, 86, 87, 89–91, 93, 94, 97, 98, 101, 102, 104, 108–110, 113, 114, 116, 118, 119, 121, 123, 124, 127, 129, 130, 132, 133, 135, 136, 143–146

DCSO DMP Common Standard Ontology. 1, 3, 21, 22, 30, 31, 74, 76, 78, 82, 83, 94, 132, 141, 145

DMP Data Management Plan. ix, xi, 1–8, 11, 18, 20, 22, 25, 27, 30–33, 37–43, 45, 48–76, 78–80, 82–84, 86, 87, 89–92, 94–102, 104–112, 114–119, 121–124, 127–133, 135, 141–146

DMPQV Data Management Plan Quality Vocabulary. 57, 58, 60, 66, 78, 84, 85, 95, 101–105, 109, 111, 114, 116–120, 122, 124, 127–131, 133, 135, 141, 144–146

DOI Digital Object Identifier. 30, 94, 95, 100, 103, 104, 110, 113, 115–117, 121, 142, 144–146

DQV Data Quality Vocabulary. 23, 24, 32, 34, 35, 57, 60, 131, 141, 143

EA Enterprise Architecture. 8, 45, 46, 48, 56, 62, 129

EOSC European Open Science Cloud. 11

FAIR Findable, Accessible, Interoperable, Reusable. 1–6, 8, 11, 12, 14, 16, 18, 20, 22, 25, 32, 34, 35, 41, 50, 52, 53, 61, 63, 74, 79, 90, 92, 94, 95, 104, 105, 108, 116, 127, 128, 132, 133, 143

HTTP Hypertext Transfer Protocol. 92, 94, 115

IRI Internationalized Resource Identifier. 85

JSON JavaScript Object Notation. 3, 28, 29, 79, 95, 99, 132, 145

KB Knowledge Base. 6

KG Knowledge Graph. 8, 81

kNN k-Nearest Neighbors. 110

LD Linked Data. 18, 20

maDMP Machine Actionable Data Management Plan. ix, xi, 1–9, 20, 21, 27–30, 37–42, 51–54, 57, 59, 62, 64, 66–69, 73, 74, 78, 81, 86, 87, 94, 95, 97, 99, 127–129, 132, 141, 143–145

MI Maturity Indicator. 11, 34

NLP Natural Language Processing. 130

NSF National Science Foundation. 1

ODRL Open Digital Rights Language. 131

ORCID Open Researcher and Contributor ID. 94

OWL Web Ontology Language. 21–23, 74, 76

PID Persistent Identifier. 29, 30, 59, 63, 76, 79, 80, 94, 95, 100, 110

PROV PROV Vocabulary. 57, 61

RDA Research Data Alliance. ix, xi, 1, 6–8, 11, 14, 16, 21, 27, 30, 34, 51, 54, 74, 127, 129, 143

RDF Resource Description Framework. 2, 21–23, 30, 65, 74, 76, 78, 84, 94, 99, 107, 131, 132, 135

RDM Research Data Management. 1, 2, 25, 32, 38, 41, 42, 48, 51, 74, 127, 131, 133

re3data Registry of Research Data Repositories. 110, 114, 144

REST Representational State Transfer. 76

RO-Crates Research Object Crates. 96

SHACL Shapes Constraint Language. 3, 21, 22, 76, 84–87, 89, 107, 108, 110, 117, 118, 130–132, 144, 145

SHEX Shape Expressions. 3, 30

SKG Scientific Knowledge Graph. 2, 5, 6, 8, 23, 25, 26, 40, 42, 63, 64, 79, 81, 100, 108, 128, 130, 131, 141, 144

SPARQL SPARQL Protocol and RDF Query Language. 3, 21–23, 76, 82–84, 90, 94, 95, 105, 131, 145

SQL Structured Query Language. 22

TOGAF The Open Group Architecture Framework. 45, 46, 48, 56, 62, 129, 141

UML Unified Modelling Language. 48, 56

URI Uniform Resource Identifier. 21, 87

URL Uniform Resource Locator. 94, 114, 115, 121, 144

W3C World Wide Web Consortium. 7, 21–23, 57, 76, 83, 131

WG Working Group. 14

Bibliography

[AFM+20] Amir Aryani, Martin Fenner, Paolo Manghi, Andrea Mannocci, and Markus Stocker. Open Science Graphs Must Interoperate! In *ADBIS, TPDL and EDA 2020 Common Workshops and Doctoral Consortium*, volume 1260, pages 195–206. Springer International Publishing, Cham, 2020.

[AGG12] Samuil Angelov, Paul Grefen, and Danny Greefhorst. A framework for analysis and design of software reference architectures. *Information and Software Technology*, 54(4):417–431, April 2012.

[AI20] Riccardo Albertoni and Antoine Isaac. Introducing the Data Quality Vocabulary (DQV). *Semantic Web*, 12(1):81–97, November 2020.

[Bal12] Alexander Ball. *Review of Data Management Lifecycle Models*. University of Bath, UK United Kingdom, February 2012.

[BLHL23] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web: A New Form of Web Content that is Meaningful to Computers will Unleash a Revolution of New Possibilities. In *Linking the World's Information: Essays on Tim Berners-Lee's Invention of the World Wide Web*, volume 52, pages 91–103. Association for Computing Machinery, New York, NY, USA, 1 edition, September 2023.

[BUD+19] Bradley W. Bishop, Judit Ungvari, Rowena I Davis, Tina Lee, Lise Goudeseune, Arika Virapongse, and Robert J. Samors. Belmont Forum Data Management Plan Scorecard. November 2019.

[Car17] Jake Carlson. An Analysis of Data Management Plans from the University of Michigan. February 2017.

[CCE+22] João Cardoso, Leyla J. Castro, Fajar J. Ekaputra, Marie C. Jacquemot, Marek Suchánek, Tomasz Miksa, and José Borbinha. DCSO: towards an ontology for machine-actionable data management plans. *Journal of Biomedical Semantics*, 13(1), 2022.

[CFIGC20] Andrea Cimmino, Alba Fernández-Izquierdo, and Raúl García-Castro. Astrea: Automatic Generation of SHACL Shapes from Ontologies. *The Semantic Web*, 12123:497–513, May 2020.

[CHWL14] Richard Cyganiak, David Hyland-Wood, and Markus Lanthaler. RDF 1.1 Concepts and Abstract Syntax. *W3C Proposed Recommendation*, January 2014.

[CJM+20] João Cardoso, Sarah Jones, Tomasz Miksa, Adil Hasan, Maria Praetzellis, Paulette Lieby, Elli Papadopoulou, and Kerstin Gierend. Mapping of madmps to funder templates, July 2020.

[CPB20] João Cardoso, Diogo Proença, and José Borbinha. Machine-Actionable Data Management Plans: A Knowledge Retrieval Approach to Automate the Assessment of Funders' Requirements. 12036:118–125, 2020.

[DH20] Anusuriya Devaraju and Robert Huber. F-UJI - An Automated FAIR Data Assessment Tool, October 2020.

[DH21] Anusuriya Devaraju and Robert Huber. An automated solution for measuring the progress toward FAIR research data. *Patterns*, 2(11):100370, November 2021.

[DLA14] Jeremy Debattista, Christoph Lange, and Sören Auer. Luzzu - A Framework for Linked Data Quality Assessment, December 2014.

[DMC+21] Anusuriya Devaraju, Mustapha Mokrane, Linas Cepinskas, Robert Huber, Patricia Herterich, Jerry De Vries, Vesa Akerman, Hervé L'Hours, Joy Davidson, and Michael Diepenbroek. From Conceptualization to Implementation: FAIR Assessment of Research Data Objects. *Data Science Journal*, 20:4, February 2021.

[Eur21] Science Europe. Practical Guide to the International Alignment of Research Data Management - Extended Edition. January 2021.

[fai] FAIRassist.org. [Accessed Nov. 14, 2023, `https://fairassist.org/`].

[FBM21] Raffael Foidl, Lea Salome Brugger, and Tomasz Miksa. Automating evaluation of machine-actionable data management plans with semantic web technologies. 2021.

[FL21] Michael Färber and David Lamprecht. The data set knowledge graph: Creating a linked open data source for data sets. *Quantitative Science Studies*, 2(4):1324–1355, December 2021.

[fRaIECBA+21] Directorate-General for Research and Innovation (European Commission), EOSC Executive Board, Jan Magnus Aronsen, Oya Beyan, Natalie Harrower, András Holl, Rob W. W. Hooft, Pedro Principe, Ana Slavec, Sarah Jones, and Françoise Genova. Recommendations on FAIR metrics for EOSC. 2021.

[GHG+13] GFZ German Research Centre For Geosciences, Humboldt-Universität Zu Berlin, Germany Karlsruhe Institute Of Technology (KIT), Purdue University Libraries, Roland Bertelmann, Matt Buys, Helena Cousijn, Uwe Dierolf, Kirsten Elger, Martin Fenner, Lea Maria Ferguson, Florian Fritze, Claudio Fuchs, Hans-Jürgen Goebelbecker, Jens Gundlach, Maxi Kindling, Gabriele Kloska, Jens Klump, Claudia Kramer, Stefka Manova, Heinz Pampel, Vivien Petras, Evelyn Reuter, Jessika Rücknagel, and Stephanie van de Sandt. Registry of Research Data Repositories (Re3Data). 2013.

[Gia16] Ronald Giachetti. *Design of Enterprise Systems*. 2016.

[GRDL+23] Alban Gaignard, Thomas Rosnet, Frédéric De Lamotte, Vincent Lefort, and Marie-Dominique Devignes. FAIR-Checker: supporting digital resource findability and reuse with Knowledge Graphs and Semantic Web standards. *Journal of Biomedical Semantics*, 14(1):7, July 2023.

[Gro20] FAIR Data Maturity Model Working Group. FAIR Data Maturity Model. Specification and Guidelines. June 2020.

[GRW12] Kate Goddard, Abdul Roudsari, and Jeremy C Wyatt. Automation bias: a systematic review of frequency, effect mediators, and mitigators. *Journal of the American Medical Informatics Association : JAMIA*, 19(1):121–127, 2012.

[GS22] Peter Hruschka Gernot Starke. arc42: Template for architecture communication, 2022.

[Hev07] Alan R Hevner. A Three Cycle View of Design Science Research. 19, 2007.

[HMPR04] Hevner, March, Park, and Ram. Design Science in Information Systems Research. *MIS Quarterly*, 28(1):75, 2004.

[HVMS19] Cynthia Hudson-Vitale and Heather Moulaison-Sandy. Data management plans: A review. *DESIDOC Journal of Library and Information Technology*, 39(6):322–328, December 2019.

[iso] ISO/IEC 25012:2008. International Standard. Accessed Nov. 07, 2023, `https://www.iso.org/standard/35736.html`.

[JPH+20] Sarah Jones, Robert Pergl, Rob Hooft, Tomasz Miksa, Robert Samors, Judit Ungvari, and Rowena Davis. Data Management Planning: How Requirements and Solutions are Beginning to Converge. *International Journal of Big Data Intelligence*, 2(1-2):208–219, 2020.

[KAN+22] N.A. Krans, A. Ammar, P. Nymark, E.L. Willighagen, M.I. Bakker, and J.T.K. Quik. FAIR assessment tools: evaluating use and performance. *NanoImpact*, 27:100402, July 2022.

[KM15] Mary Anne Kennan and Lina Markauskaite. Research Data Management Practices: A Snapshot in Time. *International Journal of Digital Curation*, 10(2), June 2015.

[KSD+20] Fabian Kirstein, Kyriakos Stefanidis, Benjamin Dittwald, Simon Dutkowski, Sebastian Urbanek, and Manfred Hauswirth. Piveau: A Large-Scale Open Data Management Platform Based on Semantic Web Technologies. In Andreas Harth, Sabrina Kirrane, Axel-Cyrille Ngonga Ngomo, Heiko Paulheim, Anisa Rula, Anna Lisa Gentile, Peter Haase, and Michael Cochez, editors, *The Semantic Web*, Lecture Notes in Computer Science, pages 648–664, Cham, 2020. Springer International Publishing.

[LSGG18] André Langer, Valentin Siegert, Christoph Göpfert, and Martin Gaedke. SemQuire - Assessing the Data Quality of Linked Open Data Sources Based on DQV. In Cesare Pautasso, Fernando Sánchez-Figueroa, Kari Systä, and Juan Manuel Murillo Rodríguez, editors, *Current Trends in Web Engineering*, Lecture Notes in Computer Science, pages 163–175, Cham, 2018. Springer International Publishing.

[MBA+] Paolo Manghi, Alessia Bardi, Claudio Atzori, Miriam Baglioni, Natalia Manola, Pedro Principe, Michele Artini, Amelie Becker, Michele De Bonis, Nikos Houssos, Katerina Iatropoulou, Antonis Lempesis, Aenne Loeden, Stefania Martziou, and Eloy Rodrigues. The OpenAIRE Research Graph Data Model. [Accessed Sept. 19, 2023, `https://graph.openaire.eu/docs/data-model/`].

[MG23] Jorge Martinez-Gil. Framework to Automatically Determine the Quality of Open Data Catalogs, August 2023.

[Mic15] William K. Michener. Ten Simple Rules for Creating a Good Data Management Plan. *PLOS Computational Biology*, 11(10):e1004525, October 2015.

[MOR22] Tomasz Miksa, Simon Oblasser, and Andreas Rauber. Automating Research Data Management Using Machine-Actionable Data Management Plans. *ACM Transactions on Management Information Systems*, 13(2):1–22, June 2022.

[MRGB17] Tomasz Miksa, Andreas Rauber, Roman Ganguly, and Paolo Budroni. Information Integration for Machine Actionable Data Management Plans. *International Journal of Digital Curation*, 12(1):22–35, September 2017.

[MSMJ19] Tomasz Miksa, Stephanie Simms, Daniel Mietchen, and Sarah Jones. Ten principles for machine-actionable data management plans. *PLOS Computational Biology*, 15(3):e1006750, March 2019.

[MSS+23] Tomasz Miksa, Marek Suchánek, Jan Slifka, Vojtech Knaisl, Fajar J. Ekaputra, Filip Kovacevic, Annisa Maulida Ningtyas, Alaa El-Ebshihy, and Robert Pergl. Towards a Toolbox for Automated Assessment of Machine-Actionable Data Management Plans. 22(1):28, August 2023.

[MVH04] Deborah L. McGuinness and Frank Van Harmelen. OWL web ontology language overview. *W3C recommendation*, 2004.

[MWH+23] Mario Moser, Jonas Werheid, Tobias Hamann, Anas Abdelrazeq, and Robert Schmitt. Which FAIR are you?: A Detailed Comparison of Existing FAIR Metrics in the Context of Research Data Management. *Proceedings of the Conference on Research Data Infrastructure*, 1, September 2023.

[MWN20] Tomasz Miksa, Paul Walk, and Peter Neish. RDA DMP Common Standard for Machine-actionable Data Management Plans. September 2020.

[MWN+21] Tomasz Miksa, Paul Walk, Peter Neish, Simon Oblasser, Hollydawn Murray, Tom Renner, Marie-Christine Jacquemot-Perbal, João Cardoso, Trond Kvamme, Maria Praetzellis, Marek Suchánek, Rob Hooft, Benjamin Faure, Hanne Moa, Adil Hasan, and Sarah Jones. Application Profile for Machine-Actionable Data Management Plans. *Data Science Journal*, 20(1):32, October 2021.

[OMK] Simon Oblasser, Tomasz Miksa, and Asanobu Kitamoto. Finding a Repository with the Help of Machine-Actionable DMPs: Opportunities and Challenges. *International Journal of Digital Curation*, 15.

[ope] The OpenAIRE Research Graph. Accessed Sept. 19, 2023, `https://www.openaire.eu/blogs/the-openaire-research-graph`.

[PHM23] Joakim Philipson, Adil Hasan, and Hanne Moa. Making Data Management Plans Machine Actionable: Templates and Tools. *Data Science Journal*, 22:29, 2023.

[PHS+19] Robert Pergl, Rob Hooft, Marek Suchánek, Vojtěch Knaisl, and Jan Slifka. "Data Stewardship Wizard": A Tool Bringing Together Researchers, Data Stewards, and Data Experts around Data Management Planning. *CODATA Data Science Journal*, 18(1):59, December 2019.

[pid] Persistent identifiers - OA Journals Toolkit. Accessed Sept. 19, 2023, https://www.oajournals-toolkit.org/infrastructure/persistent-identifiers.

[PK22] Paolo Pareti and George Konstantinidis. A Review of SHACL: From Data Validation to Schema Reasoning for RDF Graphs. In Mantas Šimkus and Ivan Varzinczak, editors, *Reasoning Web. Declarative Artificial Intelligence : 17th International Summer School 2021, Leuven, Belgium, September 8–15, 2021, Tutorial Lectures*, Lecture Notes in Computer Science, pages 115–144. Springer International Publishing, Cham, 2022.

[PMSBG23] Ngoc-Minh Pham, Heather Moulaison-Sandy, Bradley Wade Bishop, and Hannah Gunderman. Data Management Plans: Implications for Automated Analyses. *CODATA Data Science Journal*, 22(1):2, January 2023.

[RLH22] Kashif Rabbani, Matteo Lissandrini, and Katja Hose. SHACL and ShEx in the Wild: A Community Survey on Validating Shapes Generation and Adoption. In *Companion Proceedings of the Web Conference 2022*, pages 260–263, Virtual Event, Lyon France, April 2022. ACM.

[RPZ+23] Livio Robaldo, Francesco Pacenza, Jessica Zangari, Roberta Calegari, Francesco Calimeri, and Giovanni Siragusa. Efficient compliance checking of RDF data. *Journal of Logic and Computation*, 33(8):1753–1776, December 2023.

[Rum16] Bernhard Rumpe. *Modeling with UML*. Springer International Publishing, Cham, 2016.

[SAB+23] Dorothea Strecker, Alexandra Axtmann, Roland Bertelmann, Helena Cousijn, Kirsten Elger, Lea Maria Ferguson, David Fichtmüller, Catherine Jones, Iris Lindenmann, Charlotte Neidiger, Thanh Binh Nguyen, Jiban K. Pal, Heinz Pampel, Vivien Petras, Edeltraud Schnepf, Angelika Semrau, Robert Ulrich, Arne Upmeier, Paul Vierkant, Hui Wang, Gabriele Weickert, Nina Weisweiler, Sarah C. Williams, Michael Witt, and Sarah J. Wright. Metadata schema for the description of research data repositories : version 4.0. 2023.

[SJM+18] Stephanie Simms, Sarah Jones, Tomasz Miksa, Daniel Mietchen, Natasha Simons, and Kathryn Unsworth. A Landscape Survey of

ActiveDMPs. *International Journal of Digital Curation*, 13(1):204–214, 2018.

[SJMM17] Stephanie Simms, Sarah Jones, Daniel Mietchen, and Tomasz Miksa. Machine-actionable data management plans (maDMPs). *Research Ideas and Outcomes*, 3, May 2017.

[skg] Scientific Knowledge Graph. Accessed Nov. 07, 2023, `https://skg.kmi.open.ac.uk/SKG2020/`.

[SMRS+19] Susanna-Assunta Sansone, Peter McQuilton, Philippe Rocca-Serra, Alejandra Gonzalez-Beltran, Massimiliano Izzo, Allyson L. Lister, and Milo Thurston. FAIRsharing as a community approach to standards, repositories and policies. *Nature Biotechnology*, 37(4):358–367, April 2019.

[SRSC+22] Stian Soiland-Reyes, Peter Sefton, Mercè Crosas, Leyla Jael Castro, Frederik Coppens, José M. Fernández, Daniel Garijo, Björn Grüning, Marco La Rosa, Simone Leo, Eoghan Ó Carragáin, Marc Portier, Ana Trisovic, RO-Crate Community, Paul Groth, and Carole Goble. Packaging research artefacts with RO-Crate. *Data Science*, 5(2):97–138, January 2022.

[The09] The Open Group. TOGAF 9 - The Open Group Architecture Framework Version 9, 2009.

[WB13] Karl E. Wiegers and Joy Beatty. *Software Requirements*. Pearson Education, August 2013.

[WBNZ17] Mary Williams, Jacqueline Bagwell, and Meredith Nahm Zozus. Data management plans: the missing perspective. *Journal of Biomedical Informatics*, 71:130–142, July 2017.

[WCW+16] Amanda Whitmire, Jake Carlson, Brian Westra, Patricia Hswe, and Susan Wells Parham. The DART Project: using data management plans as a research tool. January 2016.

[WDA+16] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Merce Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J. G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A. C. 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E. Martone, Albert Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, 3(1):160018, March 2016.

[WDS+19] Mark D. Wilkinson, Michel Dumontier, Susanna-Assunta Sansone, Luiz Olavo Bonino da Silva Santos, Mario Prieto, Dominique Batista, Peter McQuilton, Tobias Kuhn, Philippe Rocca-Serra, Merce Crosas, and Erik Schultes. Evaluating FAIR maturity through a scalable, automated, community-governed framework. *Scientific Data*, 6(1):174, September 2019.

[WKJ+23] Bianca Wentzel, Fabian Kirstein, Torben Jastrow, Raphael Sturm, Michael Peters, and Sonja Schimmler. An Extensive Methodology and Framework for Quality Assessment of DCAT-AP Datasets. In Ida Lindgren, Csaba Csáki, Evangelos Kalampokis, Marijn Janssen, Gabriela Viale Pereira, Shefali Virkar, Efthimios Tambouris, and Anneke Zuiderwijk, editors, *Electronic Government*, Lecture Notes in Computer Science, pages 262–278, Cham, 2023. Springer Nature Switzerland.

[WO20] Simon Werner Oblasser. Entwurf einer Architektur für maschinenausführbare For-schungsdatenmanagementpläne im institutionellen Kontext, 2020. Master Thesis, TU Wien.

[WSM+22] Mark D Wilkinson, Susanna-Assunta Sansone, Grootveld Marjan, Josefine Nordling, Richard Dennis, and David Hecker. FAIR Assessment Tools: Towards an "Apples to Apples" Comparisons. December 2022.

[WW13] Nicole Gray Weiskopf and Chunhua Weng. Methods and dimensions of electronic health record data quality assessment: enabling reuse for clinical research. *Journal of the American Medical Informatics Association*, 20(1):144–151, January 2013.

[XXWT16] Wenze Xia, Zhuoming Xu, Jie Wei, and Haimei Tian. DQFIRD: Towards Data Quality-Based Filtering and Ranking of Datasets for Data Portals. In *2016 13th Web Information Systems and Applications Conference (WISA)*, pages 18–23, September 2016.

[ZBC23] Shuxin Zhang, Nirupama Benis, and Ronald Cornet. Automated approach for quality assessment of RDF resources. *BMC Medical Informatics and Decision Making*, 23(1):90, May 2023.

[Zou23] Filip Zoubek. Integrating RDM services using machine-actionable Data Management Plans, 2023. Master Thesis, TU Wien.

[ZRM+15] Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, Jens Lehmann, and Sören Auer. Quality assessment for Linked Data: A Survey. *Semantic Web*, 7:63–93, March 2015.