DSD: The Data Source Description Vocabulary

Lisa Ehrlinger[2](✉), Johannes Schrott[1,2](✉), and Wolfram Wöß[1]

[1] Johannes Kepler University Linz, Linz, Austria
{johannes.schrott,wolfram.woess}@jku.at
[2] Software Competence Center Hagenberg GmbH, Hagenberg, Austria
{lisa.ehrlinger,johannes.schrott}@scch.at

Abstract. Training machine learning models, especially in producing enterprises with numerous information systems having different data structures, requires efficient data access. Hence, standardized descriptions of data sources and their data structures are a fundamental requirement. We therefore introduce version 4.0 of the Data Source Description Vocabulary (DSD), which represents a data source in a standardized form using an ontology. We present several real-world applications where the DSD vocabulary has been applied in recent years to demonstrate its relevance. An evaluation against the FAIR principles highlights the scientific quality and potential for reuse of the DSD vocabulary.

Keywords: Data source representation · FAIR · Vocabulary · Ontology

1 Introduction

Training machine learning (ML) models [8], integration of heterogeneous data sources [5], or data quality measurement [3,4] are exemplary tasks that involve more than one data source in an organization. To merge these data sources, a standardized description of the data sources and their data structures is required. Data Source Description Vocabulary (DSD)[1] version 4.0, which enables the standardized representation of data sources and their internal structure independently of the original type of source (e.g., database management system, comma-separated values (CSV) files).

We delimit DSD from related research in Sect. 2 and describe the details of the vocabulary in Sect. 3. Sect. 4 highlights the relevance of DSD by outlining its applications in practice. The vocabulary is evaluated against the FAIR (Findability, Accessibility, Interoperability, and Reuse [12]) principles in Sect. 5.

2 Related Work

The idea of developing a standardized representation for data sources of different types is not new. Atzeni et al. [1] present a metamodel that can represent

[1] Available online: IRI: https://w3id.org/dsd; DOI: https://doi.org/10.5281/zenodo.7773861.

G. Kotsis et al. (Eds.): DEXA 2023 Workshops, CCIS 1872, pp. 3–10, 2023.
https://doi.org/10.1007/978-3-031-39689-2_1

(amongst others) relational data models, Entity-Relationship models, and object-oriented models. Candel et al. [2] propose "U-Schema", a unified metamodel that is based on the Eclipse Modeling Framework (EMF)[2] and supports the most-widely used NoSQL systems, as well as MySQL. The DSD vocabulary is different from such metamodels since it is based on the Ontology Language (OWL)[3] for building ontologies that represent data sources.

The following OWL-based vocabularies for describing the metadata of data sources [13] have been recommended by the World WideWeb Consortium (W3C):

- the Data Catalog Vocabulary (DCAT)[4], which provides terms for describing so-called "data sets" (i.e., data sources) and services to catalog them, and
- the Vocabulary of Interlinked Datasets (VoID)[5], which is specifically tailored to describe metadata of Resource Description Framework (RDF) data sets.

In contrast to DSD, both vocabularies do not cover the structure inside a data source. There are also some vocabularies that support the representation of the internal structure of a data source, like CSV on the Web (CSVW)[6] that allows describing the structure of CSV files, or the RDF Data Cube Vocabulary[7] that is suitable for multidimensional data. All of these vocabularies are dedicated to a specific data source type, while DSD is data source type independent. The Semantic Data Dictionary (SDD) has a similar objective as DSD, but only supports tabular data in its current state (Extensible Markup Language (XML) is planned in the future) [10].

Despite the same acronym, the DSD vocabulary is also different from the DSD Schema Language [9], which is an XML schema language with higher expressiveness than the XML document type declaration (DTD)[8] or XML Schema (XSD)[9].

In summary, there is no other OWL-based vocabulary than DSD that can represent data sources, independently of their type and internal structure.

3 The Data Source Description Vocabulary (DSD)

Originally, Ehrlinger and Wöß published DSD in 2015 [5]. The vocabulary is based on OWL, RDF, and RDF Schema. The core idea of DSD is to provide a terminology for representing the structure of data sources independently of their type [5]. It can be used to represent different types of data sources (e.g., relational or graph databases, document stores) and their (internal) semantics.

Based on our experience in data modeling (Entity-Relationship (ER) models, Unified Modeling Language (UML), and ontologies) and on requirements raised

[2] https://www.eclipse.org/modeling/emf/.

[3] https://www.w3.org/TR/owl2-overview/.

[4] http://www.w3.org/ns/dcat#.

[5] http://rdfs.org/ns/void#.

[6] http://www.w3.org/ns/csvw#.

[7] http://purl.org/linked-data/cube#.

[8] https://www.w3.org/TR/REC-xml/#dt-doctype.

[9] https://www.w3.org/TR/xmlschema-0/.

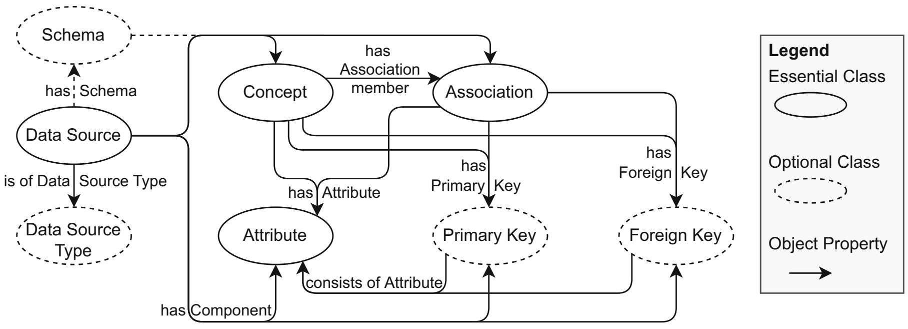

Fig. 1. OWL classes and OWL object properties in the DSD vocabulary

by company partners (cf. applications of DSD in Sect. 4), we defined a set of terms (i.e., OWL classes, object properties, and data properties) for describing data sources. Figure 1 illustrates the classes and object properties defined in DSD. For simplicity, inverse object properties are not shown. An inverse object property in OWL is a relationship between two classes where the direction of the relationship is reversed. We distinguish between "essential" classes, which are necessary for describing a data source using DSD, and "optional" classes, which provide additional non-necessary features. Below, we describe each class, in order of importance.

Essentials

- **Data Source.** A generic class for representing data sources. *Example:* A `dsd:DataSource` can represent structured data such as relational databases, semi-structured data like XML files, or NoSQL databases such as graph databases or wide-column stores.
- **Concept.** A representation of a structural part of a data source. *Example:* A `dsd:Concept` can represent a table or a view of a relational database or a class in object-oriented structures.
- **Attribute.** A `dsd:Attribute` describes a property of a `dsd:Concept`. DSD also provides OWL data properties to define certain attribute characteristics, such as, nullable or unique. *Example:* If a `dsd:Concept` represents a relational table, its attributes correspond to the columns.
- **Association.** A `dsd:Association` describes a relationship between two instances of `dsd:Concept`. There are three disjoint `dsd:Association` subclasses for aggregation, inheritance, and reference associations. For further details and also for object properties of the subclasses, we refer to [5].

Optionals

- **Schema.** Instances of `dsd:Schema` create an optional hierarchy level between data sources (instances of `dsd:DataSource`) and concepts (instances of `dsd:Concept`). Schemas allow the grouping of concepts and are commonly used in enterprise databases.

- **Data Source Type.** This class provides instances of the most common data source types, which can be assigned to instance of `dsd:DataSource`.
- **Primary Key** and **Foreign Key.** Instances of these two classes are assigned to a `dsd:Association` or `dsd:Concept` and consist of one or more instances of `dsd:Attribute` (i.e., can be composite keys).

4 Use Cases and Applications of DSD

In recent years, DSD has been used in various applications. This section discusses three areas where DSD can be useful for both researchers and practitioners.

Schema Matching and Schema Similarity. A key advantage of DSD is to make data sources and their schemas comparable. Thus, in [6], DSD was used to generate homogeneous representations of data source schemas, which could then be compared directly. The similarity of these schemas (i.e., their degree of overlap) was used as input for a metric to assess the schema quality [6].

Metadata Management. The implementation of a corporate metadata management system (e.g., a data catalog) requires comparability of data source schemas from different types. For that purpose, we employed DSD to represent different data sources in a producing company [11]. In this project, DSD was the basis to describe data sources and their internal structure, which can then be annotated with different kinds of metadata, e.g., access security metadata or the assignment of data responsibility roles.

Data Quality. In real-world scenarios, data quality assessment should be carried out on multiple (heterogeneous) data sources. Thus, the data quality tools QualIe [4] and DQ-MeeRKat [3], which aim to be data source type independent, implement connectors[10] that map the original schema of a data source to a DSD representation (see Table 1 in [5]). After calculating different data quality metrics, the measurement results can be annotated to these representations.

5 Evaluation Against the FAIR Principles

The FAIR principles define a measurable set of guidelines to assess the FAIRness of a data asset [12] and are therefore well suited to evaluate the quality (i.e., findability, accessibility, interoperability, and reuse) of DSD. We conducted a twofold evaluation: (1) an automated evaluation using FOOPS![11] in Sect. 5.1 and (2) a manual evaluation with the FAIR principles published online in Sect. 5.2.

[10] See the "connectors" Java package in https://github.com/lisehr/dq-meerkat.
[11] https://w3id.org/foops/.

5.1 Automatic Evaluation

For the automatic evaluation, we used the tool FOOPS! (Ontology Pitfall Scanner for FAIR) [7]. FOOPS! determines FAIRness by checking if Internationalized Resource Identifiers (IRIs) are resolvable and permanent, and if certain OWL properties (e.g., author, publication date, provenance information) are present.

In the automatic evaluation, DSD achieves a FAIRness score of 88%. FOOPS! does not assess DSD to be fully FAIR since it does not recognize some specific metadata. As an example, information on authors and contributors of DSD is included as instances of `foaf:Person`, but FOOPS! expects the presence of literal values.

5.2 Manual Evaluation

For each FAIR principle[12], we manually assessed and justified if it is fulfilled by DSD, as shown in detail in Table 1. Overall, we consider DSD to be fully FAIR.

Table 1. Manual evaluation against the FAIR Principles.

FAIR principle	Fulfillment	Justification
Findable		
F1. (Meta)data are assigned a globally unique and persistent identifier.	✓	The base IRI of DSD is https://w3id.org/dsd, which is unique and a persistent identifier.
F2. Data are described with rich metadata (defined by R1 below)	✓	*See detailed principles R1.1-1.3.*
F3. Metadata clearly and explicitly include the identifier of the data they describe	✓	The metadata of the vocabulary is annotated using RDF. Data (= subject) is annotated with specific (= predicate) metadata (= object).
F4. (Meta)data are registered or indexed in a searchable resource	✓	DSD is indexed in Linked Open Vocabularies (LOV)[a].
Accessible		
A1. (Meta)data are retrievable by their identifier using a standardised communications protocol	✓	The vocabulary is available online (see Footnote 1) and can be retrieved using the HTTPS protocol.
A1.1 The protocol is open, free, and universally implementable	✓	HTTPS fulfills all these criteria.
A1.2 The protocol allows for an authentication and authorisation procedure, where necessary	✓	HTTPS allows, e.g., basic-auth. In the case of DSD, no authentication and authorization are needed.

(*continued*)

[12] The FAIR principles and the corresponding descriptions in the leftmost column of Table 1 are directly taken from the GO-FAIR website (https://www.go-fair.org/fair-principles/).

Table 1. (*continued*)

FAIR principle	Fulfillment	Justification
A2. Metadata are accessible, even when the data are no longer available	✓	DSD has a DOI and is indexed in LOV[a] as well as prefix.cc[b]. Furthermore, a GitHub repository[c] exists.
Interoperable		
I1. (Meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation	✓	DSD is available online in Turtle[d] syntax.
I2. (Meta)data use vocabularies that follow FAIR principles	✓	DSD is based on RDF and OWL. It does not import any other vocabularies.
I3. (Meta)data include qualified references to other (meta)data	✓	The metadata of DSD is encoded using RDF, thus all references are qualified.
Reusable		
R1. (Meta)data are richly described with a plurality of accurate and relevant attributes	✓	*See detailed principles R1.1-1.3*
R1.1. (Meta)data are released with a clear and accessible data usage license	✓	DSD is licensed under the GNU Lesser General Public Licens (LGPL)[e].
R1.2. (Meta)data are associated with detailed provenance	✓	Provenance information is provided via DSDs GitHub repository[c]. To maintain a clear scope of the vocabulary, we do not include provenance information directly in the vocabulary.
R1.3. (Meta)data meet domain-relevant community standards	✓	The vocabulary uses RDF and OWL. Metadata information of the vocabulary is encoded with terms that are recommended as "best-practice" by FOOPS! and PyLODE[f].

[a] https://lov.linkeddata.es/dataset/lov/
[b] https://prefix.cc/
[c] https://github.com/FAW-JKU/dsd-vocabulary
[d] https://www.w3.org/TR/2014/REC-turtle-20140225/
[e] https://www.gnu.org/licenses/old-licenses/lgpl-2.1.html
[f] https://github.com/RDFLib/pyLODE

6 Conclusion and Outlook on Future Work

Although the focus of DSD is on the description of data sources, previous versions contained, e.g., a class `Stakeholder`, which was used for modelling people and their permissions to data sources. In the newest version 4.0, we removed all capabilities that do not support the core idea of DSD and suggest the reuse and

combination with other vocabularies to annotate different kinds of *metadata* to a data source. An example is the Data Quality Vocabulary (DQV)[13], which is specifically designed to represent data quality metadata. DSD 4.0 is the first version that includes a rich set of metadata as well as a permanent identifier, and thus fulfills the FAIR principles. Due to intensively using DSD in data quality tools (cf. [3,4]), we will further investigate the integration of DSD with DQV in our ongoing research. At this point, we would like to encourage other research groups to investigate the integration of additional vocabularies for annotating *metadata* to DSD data sources, e.g., security or provenance metadata.
All links in this publication were last visited on June 1, 2023.

Acknowledgements. This research has been partially funded by BMK, BMAW, and the State of Upper Austria in the frame of the SCCH competence center INTEGRATE (FFG grant no. 892418) part of the FFG COMET Competence Centers for Excellent Technologies Programme and by the "ICT of the Future" project QuanTD (no. 898626).

References

1. Atzeni, P., Gianforme, G., Cappellari, P.: A universal metamodel and its dictionary. In: Hameurlain, A., Küng, J., Wagner, R. (eds.) Transactions on Large-Scale Data- and Knowledge-Centered Systems I. LNCS, vol. 5740, pp. 38–62. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03722-1_2
2. Candel, C.J.F., Sevilla Ruiz, D., García-Molina, J.J.: A Unified Metamodel for NoSQL and Relational Databases. Information Syst. **104**, 101898 (2022). https://doi.org/10.1016/j.is.2021.101898
3. Ehrlinger, L., Gindlhumer, A., Huber, L., Wöß, W.: DQ-MeeRKat: automating Data Quality Monitoring with a Reference-Data-Profile-Annotated Knowledge Graph. In: Proceedings of the 10th International Conference on Data Science, Technology and Applications - DATA, pp. 215–222. SciTePress (2021)
4. Ehrlinger, L., Werth, B., Wöß, W.: Automated continuous data quality measurement with QuaIIe. Int. J. Adv. Softw. **11**(3 & 4), 400–417 (2018)
5. Ehrlinger, L., Wöß, W.: Semi-automatically generated hybrid ontologies for information integration. In: Joint Proceedings of the Posters and Demos Track of 11th International Conference on Semantic Systems - SEMANTiCS2015 and 1st Workshop on Data Science: Methods, Technology and Applications (DSci15), vol. 1481, pp. 100–104. CEUR Workshop Proceedings (2015). https://ceur-ws.org/Vol-1481/paper30.pdf
6. Ehrlinger, L., Wöß, W.: Automated schema quality measurement in large-scale information systems. In: Hacid, H., Sheng, Q.Z., Yoshida, T., Sarkheyli, A., Zhou, R. (eds.) QUAT 2018. LNCS, vol. 11235, pp. 16–31. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-19143-6_2
7. Garijo, D., Corcho, O., Poveda-Villalón, M.: FOOPS!: an ontology pitfall scanner for the FAIR principles. In: International Semantic Web Conference (ISWC) 2021. CEUR Workshop Proceedings, vol. 2980 (2021). http://ceur-ws.org/Vol-2980/paper321.pdf

[13] http://www.w3.org/ns/dqv#.

8. Gebru, T.: Datasheets for datasets. Commun. ACM **64**(12), 86–92 (2021). https://doi.org/10.1145/3458723
9. Klarlund, N., Møller, A., Schwartzbach, M.I.: The DSD schema language. Autom. Softw. Eng. **9**, 285–319 (2002). https://doi.org/10.1023/A:1016376608070
10. Rashid, S.M., et al.: The semantic data dictionary - an approach for describing and annotating data. Data Intell. **2**(4), 443–486 (2020). https://doi.org/10.1162/dint_a_00058
11. Schrott, J., Weidinger, S., Tiefengrabner, M., Lettner, C., Wöß, W., Ehrlinger, L.: GOLDCASE: a generic ontology layer for data catalog semantics. In: Garoufallou, E., Vlachidis, A. (eds.) MTSR 2022. CCIS, vol. 1789, pp. 26–38. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-39141-5_3
12. Wilkinson, M.D., et al.: The FAIR Guiding Principles for scientific data management and stewardship. Sci. Data **3**(1), 160018 (2016). https://doi.org/10.1038/sdata.2016.18
13. World Wide Web Consortium: All Standards and Drafts - W3C. https://www.w3.org/TR/. Accessed 21 Feb 2023