Universität Leipzig
Fakultät für Mathematik und Informatik
Institut für Informatik
Abteilung für betriebliche Informationssysteme

Master Thesis

DataID

Semantically Rich Metadata for Complex Datasets

Abstract: <Gegenstand und Resultate der Arbeit. Was ist neu? Warum sollte man die Arbeit lesen?>

Leipzig, January 2017

Markus Freudenberg
freudenberg@informatik.uni-leipzig.de

Betreuender Hochschullehrer:

Dr.-Ing. Sebastian Hellmann and
Dimitris Kontokostas
Fakultät für Mathematik und Informatik
Betriebliche Informationssysteme, Semantic Web

Acknowledgements

Contents

List of Figures

Namespaces

A list of namespaces and their prefixes used throughout this work:

Prefix	Namespace
dataid	`http://dataid.dbpedia.org/ns/core#`
dataid-ld	`http://dataid.dbpedia.org/ns/ld#`
datacite	`http://purl.org/spar/datacite/`
dcat	`http://www.w3.org/ns/dcat#`
dct	`http://purl.org/dc/terms/`
dlo	`http://aligned-project.eu/ontologies/dlo#`
foaf	`http://xmlns.com/foaf/0.1/`
lvont	`http://lexvo.org/ontology#`
odrl	`http://www.w3.org/ns/odrl/2/`
org	`http://www.w3.org/ns/org#`
owl	`http://www.w3.org/2002/07/owl#`
prov	`http://www.w3.org/ns/prov#`
rdf	`http://www.w3.org/1999/02/22-rdf-syntax-ns#`
rdfs	`http://www.w3.org/2000/01/rdf-schema#`
r3d	`http://www.re3data.org/schema/3-0#`
sd	`http://www.w3.org/ns/sparql-service-description#`
skos	`http://www.w3.org/2004/02/skos/core#`
spdx	`http://spdx.org/rdf/terms/#`
time	`http://www.w3.org/2006/time#`
void	`http://rdfs.org/ns/void#`
xsd	`http://www.w3.org/2001/XMLSchema#`

Disambiguation

There are multiple interpretations of the word/acronym **`DataID`** depending on the context. It can refer to a **`DataID`** metadata document, the serialisation of a **`DataID`** RDF graph. Such a graph is the result of the appliance of **`DataID`** ontologies to one or more datasets, resulting in a collection of RDF statements based on these ontologies. Or it is used to name an instance of the concept `dataid:DataId`, meaning the entry into a `dcat:Catalog`, the most abstract entity in every **`DataID`**. I will be explicit and use the terms **`DataID`** document, **`DataID`** graph or **`DataID`** resource (or instance, entity) in the remainder of this thesis.

add blank pages

1 Introduction

1.1 Motivation

In 2006, Clive Humby coined the phrase "the new oil" for (digital) data[1], heralding the ever-expanding realm of what is now summarised as Big Data. Attributed with the same transformative and wealth-producing abilities, once connected to crude oil bursting out of the earth, data has become a cornerstone of economical and societal visions. In fact, the amount of data generated around the world has increased dramatically over the last years, begging the question if those visions have already come to pass.

The steep increase in data produced can be ascribed to multiple factors. To name just a few:

- The growth in content and reach of the World Wide Web[2].
- The digitalising of former analogue data (e.g. [3], [4])
- The realisation of what is called the Internet of Things (IoT)[5].
- The shift of classic fields of research and industry to computer-aided processes and digital resource management (e.g. digital humanities[6], industry 4.0[7]).
- Huge data collections about protein sequences or human disease taxonomies are established in the life sciences[8].
- Research areas like Natural Language Processing or Machine Learning are generating and refining data[9].

[1] https://www.theguardian.com/technology/2013/aug/23/tech-giants-data
[2] http://www.internetworldstats.com/emarketing.htm
[3] https://www.loc.gov/programs/national-recording-preservation-board/
[4] https://archive.org
[5] http://siliconangle.com/blog/2015/10/28/page/3\#post-254300
[6] http://www.dh.uni-leipzig.de/wo/
[7] http://www.europarl.europa.eu/RegData/etudes/STUD/2016/570007/IPOL_STU(2016)570007_EN.pdf
[8] https://www.ncbi.nlm.nih.gov/genbank/statistics/
[9] http://archive.ics.uci.edu/ml/

- In addition, open data initiatives like the Open Knowledge Foundation[10] are following the call for 'Raw data, Now!'[11] of Tim Berners-Lee, demanding open data from governments and organisations.

But with Big Data comes a big challenge. The increasing deluge of data is submerging data producers and possible consumers in a wave of unfiltered, unstructured and apparently unmanageable information. As a new discipline, data engineering is dealing with the fallout of this trend, namely with issues of how to extract, aggregate, store, refine, combine and distribute data of different sources in ways which give equal consideration to the four V's of Big Data: Volume, Velocity, Variety and Veracity[12].

Datasets are the building blocks of these endeavours. They are the combination of multiple data points (datums) bundled together by at least one dimension of distinction (such as source, topic or temporal information). When working with these chunks of data, extra data about data (or metadata) is needed. Dataset metadata enables users to discover, understand and (automatically) process the data it holds, as well as providing provenance on how a dataset came into existence. This metadata is often created, maintained and stored in diverse data repositories featuring disparate data models that are often unable to provide the metadata necessary to automatically process the datasets described. In addition, many use cases for dataset metadata call for more specific information than provided by most available metadata vocabularies. Extending existing metadata models to fit these scenarios is a cumbersome process resulting often in non-reusable solutions.

One vocabulary for dataset metadata is breaking this trend. Since its introduction in 2013, the Data Catalog Vocabulary [1], has been widely adopted as a foundation for dataset metadata in research, government and industry[13]. The very general approach adopted by the authors of DCAT allows for portraying any given (digital) object with this ontology. Extending DCAT is very easy and mappings to other metadata formats are not difficult to achieve.

Conversely, the general approach of DCAT is often too imprecise where specificity is needed, resulting in:

- Insufficient provenance information
- Missing relations between Datasets
- Relations to agents are too cursory

[10] https://okfn.org
[11] http://www.wired.co.uk/news/archive/2012-11/09/raw-data
[12] http://www.ibmbigdatahub.com/infographic/four-vs-big-data
[13] https://joinup.ec.europa.eu/sites/default/files/isa_field_path/2016-05-13_dcat-ap_intro_v0.05.pdf

- Technical description of resources on the Web (e.g. API endpoints) is lacking, restricting the accessibility of the data
- General lack of specificity, inviting non-machine-readable expressions of resources

Similar findings were concluded at the W3C/VRE4EIC workshop 'Smart Descriptions & Smarter Vocabularies' (SDSVoc) in 2016 [2].

add dbpedia

As a result of lacking specificity, current representations of datasets with DCAT are often not contributing to the main benefits of publishing data on the Web: *"Reuse, Comprehension, Linkability, Discoverability, Trust, Access, Interoperability and Processability"* [3]. This, in turn, amplifies broader problems with published datasets, especially in the open data community, reflected by the Open Data Strategy[14], defining the following six barriers for "open public data" [15], proposed by the European Commission in 2011:

1. a lack of information that certain data actually exists and is available,
2. a lack of clarity of which public authority holds the data,
3. a lack of clarity about the terms of re-use,
4. data made available in formats that are difficult or expensive to use,
5. complicated licensing procedures or prohibitive fees,
6. exclusive re-use agreements with one commercial actor or re-use restricted to a government-owned company.

Many issues with DCAT itself or their manifestation in reality can be solved by existing ontologies, even when restricted only to W3C recommended ontologies. For example, the PROV Ontology [4], deals with questions on how to record provenance information on a very granular level. While the Open Digital Rights Language [5] provides machine readable descriptions of licenses and other policies. The existence of problems, like those listed above, despite these offered solutions, speaks to a larger problem of missing organisational structures for landscaping of vocabularies (offering recommendations on combining, revising and usage of ontologies). A study of 91 commonly used vocabularies concluded:

explain what that means

> "Our validation detected a total of 6 typos, 14 missing or unavailable ontologies, 73 language level errors, 310 instances of ontology namespace violations and 2 class cycles which we believe to be errors." [6]

[14] http://europa.eu/rapid/press-release_IP-11-1524_en.htm?locale=en
[15] http://europa.eu/rapid/press-release_MEMO-11-891_en.htm

These errors accumulate when strong interdependencies exist between vocabularies, adding logical and practical problems and aggravating unification issues of ontologies.

1.2 Objectives

In this thesis, I will present the metadata model of **DataID**, a multi-layered metadata ecosystem, which, in its core, describes complex datasets and their different manifestations, their relations to other datasets and agents (such as persons or organisations) endowed with rights and responsibilities.

Improving the portrayal of **provenance**, **licensing** and **access**, while maintaining the easy **extensibility** and **interoperability** of DCAT, are the linchpin objectives in my effort to present a comprehensive, extensible and interoperable metadata vocabulary. Multiple well established ontologies (such as PROV-O, VOID and FOAF) are reused for maximum compatibility to establish a uniform and accepted way to describe and deliver dataset metadata for arbitrary datasets and to put existing standards into practice.

The **DataID Ecosystem** is a suite of ontologies comprised of **DataID core** and multiple extension ontologies, clustered around **DataID core**. It is the result of a modularisation process, which was necessary to preserve **extensibility** and **interoperability** of the DCAT vocabulary, on which all ontologies are based.

I want to present my solution for most of the current problems with dataset metadata in general and DCAT in particular, following these objectives:

1. Provide sufficient support for extensive and machine-readable representations for **provenance**, **licensing** and data **access**.
2. Extend DCAT with well-established ontologies to resolve the discussed issues.
3. Show, that by modularising into a landscape of ontologies, **DataID** preserves the general character of DCAT, supporting **extensibility** and **interoperability**.
4. Prove that the resulting ecosystem is capable of serving for complex demands on dataset metadata (proving **extensibility**).
5. Demonstrate the **interoperability** with other metadata formats.

6. Evaluate the universal applicability of **DataID** for datasets against common demands on data publications.

In addition, **DataID** shall support the FAIR Data principles [7] (section 2.1.5) as well as the best practices defined by the Data on the Web Best Practices working group [3] of the W3C (restricted to those practices where metadata is of concern).

DataID was developed under the sponsorship of the H2020 project ALIGNED[16] (GA-644055), following its main goals:

- to be part of a unified software and data engineering process;
- describing the complete data lifecycle and domain model;
- with an emphasis on quality, productivity and agility.

In the context of ALIGNED, **DataID** is part of a shared model of software and data engineering to enable unified governance and coordination between aligned co-evolving software and data lifecycles:

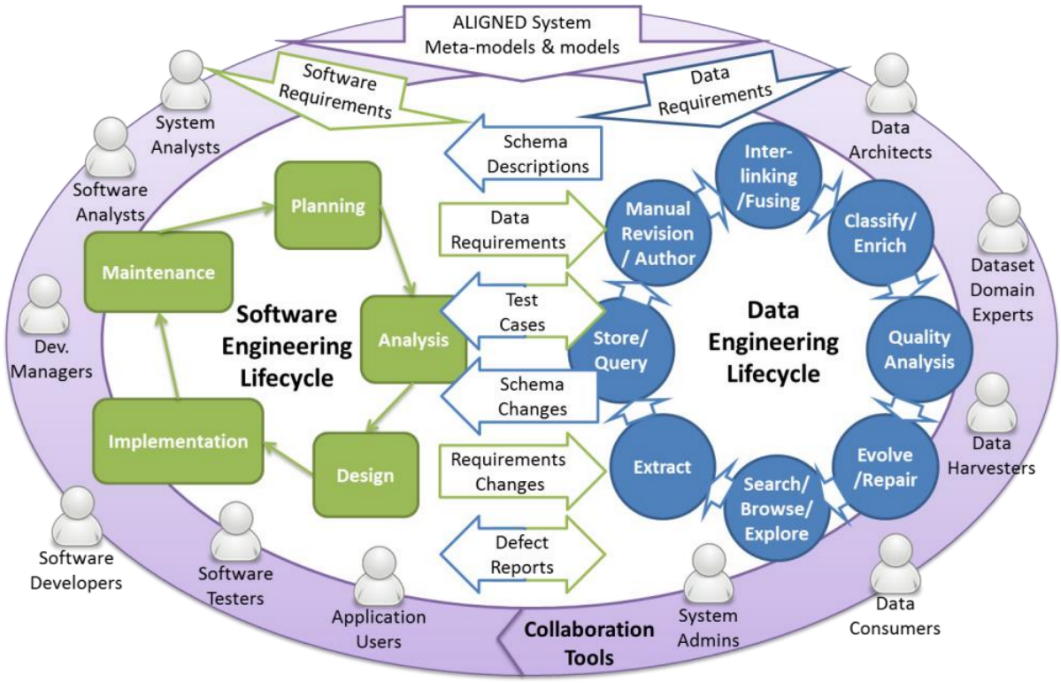


Figure 1.1: ALIGNED Software and Data Engineering Processes

add a last sentence?

[16] http://aligned-project.eu

1.3 Structure

This work is a comprehensive introduction to **DataID**, with a particular focus on the **DataID core** ontology, at the heart of the **DataID Ecosystem**. It is largely based on four publications:

1. Martin Brümmer, Ciro Baron, Ivan Ermilov, Markus Freudenberg, Dimitris Kontokostas and Sebastian Hellmann "DataID: Towards Semantically Rich Metadata for Complex Datasets". In: Proceedings of the 10th International Conference on Semantic Systems. SEM '14. Leipzig, Germany: ACM, 2014, pp. 84–91. [8] An introduction to the first version of **DataID**.
2. Monika Solanki, Bojan Bozic, Markus Freudenberg, Dimitris Kontokostas, Christian Dirschl and Rob Brennan "Enabling Combined Software and Data Engineering at Web-Scale: The ALIGNED Suite of Ontologies". In: The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part II. 2016, pp. 195–203. [9]: An overview of the landscape of ontologies developed by the ALIGNED project.
3. Markus Freudenberg, Martin Brümmer, Jessika Rücknagel, Robert Ulrich, Thomas Eckart, Dimitris Kontokostas and Sebastian Hellmann "The Metadata Ecosystem of DataID". In: Metadata and Semantics Research: 10th International Conference, MTSR 2016, Göttingen, Germany, November 22-25, 2016, Proceedings. Ed. by Emmanouel Garoufallou et al. Cham: Springer International Publishing, 2016, pp. 317–332. I S B N : 978-3-319-49157-8. [10]: An introduction the **DataID Ecosystem**
4. DataID core Ontology (2017): A W3C member submission of the University of Leipzig, under review by the W3C at the time of writing, authored by Martin Brümmer and me.

add ref

After a look at related work (chapter 3) on the subject of dataset metadata, I will present the **DataID Ecosystem** in chapter 4, to introduce the guiding principles of this work. Chapter 5 describes the **DataID core** ontology in detail, containing a running example of a DBpedia language edition. Chapter 6 provides a best practice about publishing data on the Web with **DataID**, followed by an application of those practice to a real example on how to solve complex metadata challenges with the **DataID Ecosystem**, by looking at Data Management Plans (chapter 7). Chapter 8 provides mappings between **DataID** and multiple CMD profiles of the Component MetaData Infrastructure (CMDI). I will evaluate **DataID** in chapter 9 and discuss its development and future work in chapter 10.

what is this?

2 Foundations

2.1 Data

Data is an almost intangible term. It is highly ambiguous and touches many fields of interest, stretching from philosophy to digital signal processing. Even in the context of Information Science, Data has multiple possible definitions. Here are some of them:

> "Data is a symbol set that is quantified and/or qualified." (Prof. Aldo de Albuquerque Barreto, Brazilian Institute for Information in Science and Technology, Brazil [11])
>
> "Data are sensory stimuli that we perceive through our senses." (Prof. Shifra Baruchson–Arbib, Bar Ilan University, Israel [11])
>
> "By data, we mean known facts that can be recorded and that have implicit meaning." (Prof. Shamkant Navathe, College of Computing at the Georgia Institute of Technology, USA [12])
>
> "Etymologically, data [...] is the plural of datum, a noun formed from the past participle of the Latin verb dare–to give. Originally, data were things that were given (accepted as "true"). A data element, d, is the smallest thing which can be recognized as a discrete element of that class of things named by a specific attribute, for a given unit of measure with a given precision of measurement." (Prof. Charles H. Davis, Indiana University, USA [11])
>
> "Data are the basic individual items of numeric or other information, garnered through observation; but in themselves, without context, they are devoid of information." (Dr. Quentin L. Burrell, Isle of Man International Business School, Isle of Man [11])

Information and Data seem to be closely linked and are often used interchangeably, yet they are not the same thing:

> "Datum is every thing or every unit that could increase the human knowledge or could allow to enlarge our field of scientific, theoreti-

> cal or practical knowledge, and that can be recorded, on whichever support, or orally handed. Data can arouse information and knowledge in our mind." (Prof. Maria Teresa Biagetti, University of Rome 1, Italy; based on C. S. Peirce, 1931, 1958 [11])

I will take a broader look at the term Data, delineating it from the concepts of Information and Knowledge.

2.1.1 Data, Information, Knowledge

Data is the result of the application of syntactical rules against a set (sequence, string etc.) of signs or signals, out of which a message between a sender and recipient is constructed [13]. Additional schematics might apply, adding structural data. *Information* can be gleaned from data if one can find meaning in it. The result of this semantic expansion (or interpretation) of *Data* can be weighted by its novelty value or exceptionalness in the context of existing information, using Shannon's entropy [14]. *Data* becomes tokens of perceptions, or more commonly used in Information Science: instances of concepts, capable of changing the understanding of a context for the recipient of the original message. Linking *Information* in a context to determine their interrelations and inferring additional information under the presumption of an intellectual goal, are the processes (*Pragmatics*) turning *Information* into *Knowledge*.

The pyramidal structure depicted (fig. 2.1) is often crowned by an additional field called "Wisdom" [15]. Wisdom could be described as a form of evaluated *Knowledge* or understanding, as the pinnacle of human endeavour or enlightenment. But in the era of fake news, pathological mistrust and truthiness[17], this last step does not seem to follow naturally.

check if we need quotation here

Many authors describe this hierarchy or derivations of it. The following common aspects are presented throughout this literature [15]:

- the key elements are data, information and knowledge,
- these key elements are virtually always arranged in the same order, some models offer additional stages, such as wisdom, or enlightenment,
- the higher elements in the hierarchy can be explained in terms of the lower elements by identifying an appropriate transformation process,
- the implicit challenge is to understand and explain how data is transformed into information and information is transformed into knowledge.

[17] https://en.wikipedia.org/wiki/Truthiness

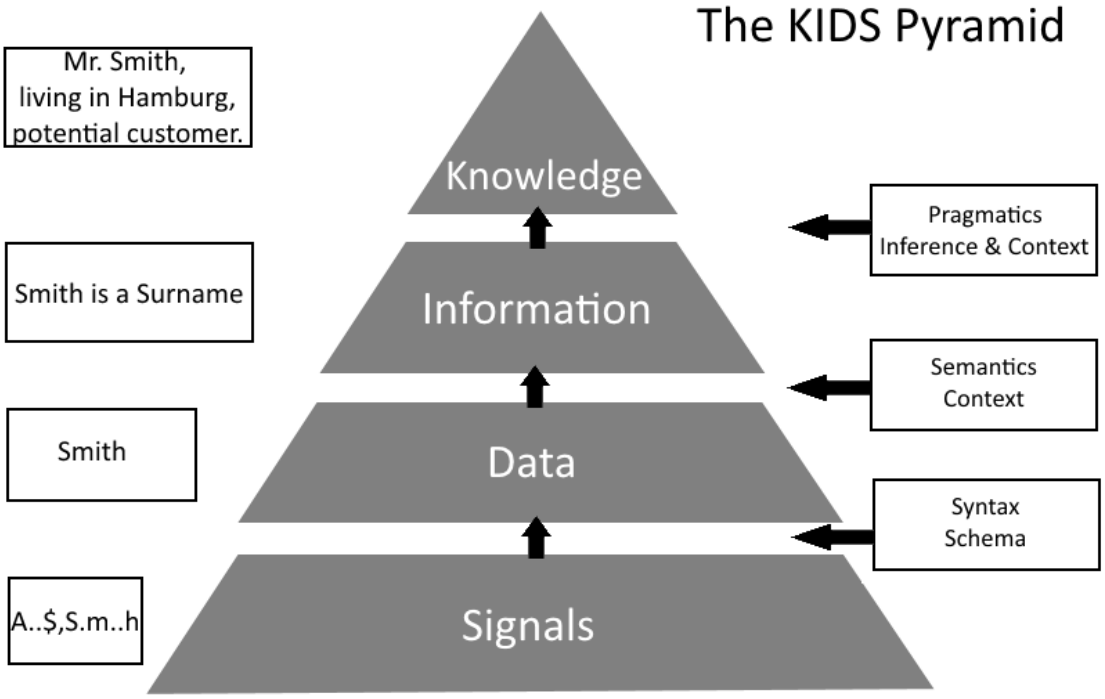


Figure 2.1: The KIDS Pyramid [13]

2.1.2 Digital Data

Digital data is represented using the binary number system of ones (1) and zeros (0). Typically, these are combined into eight of their kind - named Byte. Bytes are used to identify characters of a given alphabet, which, in turn, provide the building stones for any operation, program or data point (datum).

In general, the term Digital Data is used to describes a collection of bytes, representing a digital mapping of an analogue counterpart (e.g. sound waves), or characters of an alphabet understood by humans or programs.

Digital Data is often categorized in structured and unstructured data. Unstructured data does not follow any predefined model and has to be interpreted by the recipient (reader) by its own merit (usually free text). Structured data is strictly adherent to a given data model, facilitating its interpretation by machines and humans alike (such as [please insert]).

2.1.3 Dataset

A dataset[18] is a bundle of data, which have at least one common dimension of distinction. For example, a music album of an artist can be viewed as a dataset, where a single song represents a unit of data. Multiple songs are collected in an album with the common feature (among others) - the artist. Most commonly a dataset corresponds to a collection of structured (digital) data in a single location (e.g. a database table, XML document etc.).

Datasets can manifest in different formats (e.g. in different file types). Therefore, the distinction between dataset as a container for collecting data points of similar content or structure, and its final manifestation on a file system, is advisable.

2.1.4 Metadata

The National Information Standards Organization[19] (NISO), a United States non-profit standards organisation, published a paper in 2004, defining metadata in a widely adopted manner:

> "Metadata is structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use, or manage an information resource. Metadata is often called data about data or information about information." [16]

Metadata does not contributes additional content to the original message, but it can ease its transmission, procession and understanding. The meaning of the term metadata and for what kind of data it applies is different, depending on context, disciplines and communities. For example, library catalogue information about a certain book might be understood as metadata in regard to the book itself, while in the context of a Library Management System this type of metadata is considered data.

Specialized types of metadata can be broadly separated into three types [16]:

- **Descriptive metadata** describes a resource for purposes such as discovery and identification. It can include elements such as title, abstract, author, and keywords.
- **Structural metadata** indicates how compound objects are put together, for example, how pages are ordered to form chapters.

[18] or 'data set', though this spelling seems to be replaced more and more
[19] `http://www.niso.org/home/`

- **Administrative metadata** provides information to help manage a resource, such as when and how it was created, file type and other technical information, and who can access it. Subsets of administrative metadata include:
  - **Rights management metadata**, which deals with intellectual property rights and licenses.
  - **Preservation metadata**, which contains information needed to archive and preserve a resource.

A metadata record conforms to a given schema, since the use of unstructured data to qualify a different data resource is an exercise in futility. Various metadata schemata or ontologies are available, often describing similar types of data. The most commonly used metadata vocabulary is Dublin Core[20] (DC) by the Dublin Core Metadata Initiative (DCMI):

> "The original objective of the Dublin Core was to define a set of elements that could be used by authors to describe their own Web resources. [...] the goal was to define a few elements and some simple rules that could be applied by noncatalogers [sic]." [16]

The following example illustrates the use of Dublin Core attributes (e.g. `dc:description`) to describe a publication released as an PDF file:

```
dc:title="Metadata Demystified"
dc:creator="Brand, Amy"
dc:creator="Daly, Frank"
dc:creator="Meyers, Barbara"
dc:subject="metadata"
dc:description="Presents an overview of metadata conventions."
dc:publisher="NISO Press"
dc:publisher="The Sheridan Press"
dc:date="2003-07-01"
dc:type="Text"
dc:format="application/pdf"
dc:identifier="http://www.niso.org/standards/resources/metadata.pdf"
dc:language="en"
```

Listing 2.1: Dublin Core example

All Dublin Core attributes are optional, repeatable (non-functional) and present without order. Controlled vocabularies[21] are recommended to be used in connection with some fields (such as `dc:subject`). Since its introduction in 1995, the initial list of 15 attributes has been revised and extended, forming the so called DCMI Metadata Terms[22] (DCT). The very general character of this vo-

[20] `http://dublincore.org/documents/dces/`
[21] a set of predefined values, controlled by an institution or a body of experts
[22] `http://dublincore.org/documents/dcmi-terms/`

cabulary provides a useful foundation for more complex schemata reusing DC (for example DCAT).

Metadata can describe any resource in any state or aggregation (single resource, collections, a part of a resource). Any resources, at any abstraction level of a domain model can be substantiated with metadata. The International Federation of Library Associations and Institutions[23] (IFLA) defined the "Functional Requirements for Bibliographic Records" [17], a conceptual model for retrieval and access in online library catalogues and bibliographic databases: **Item** is an exemplar of a **Manifestation**, which embodies an **Expression**, realising a **Work** of an author.

For example, a metadata record could describe a report, a particular edition of the report, or a specific copy of that edition of the report.

is this a quote?

Metadata can be embedded in the described digital object, alongside the object (e.g. in the same directory) or stored separately. HTML documents often keep metadata in the HTML header or completely emerged in the document (see RDFa [18]). While a close coupling between data and metadata is useful when updating them together, a separate approach often simplifies the management of large numbers of records.

The advantages of reliable metadata for digital objects are manifold. These are the more poignant attributions of metadata:

Resource Discovery: Metadata is the foremost source of information which aids agents (persons, software etc.) to discover wanted data.

Organising Electronic Resources: Entities (e.g. books in a library) are organised in catalogues with the help of their digital metadata records.

Interoperability: The interoperability of two digital resources can be determined by comparing their metadata entries, on a syntactical as well as on a semantic level.

Digital Identification: Digital identifiers (such as URIs section 2.2) are stored with metadata records.

Provenance: An extensive record on provenance is a key for trustworthiness, detailing facts like source data, responsible agents or origin activities.

Quality: The quality of a data source is also established by the quality of its metadata.

Reusability: Increasing discoverability, interoperability, provenance and quality are instrumental requirements for increasing reusability.

[23] http://www.ifla.org

Preservation: "Metadata is key to ensuring that resources will survive and continue to be accessible into the future." [16]

In the chain of processes transforming Signals, Data, Information into Knowledge (section 2.1.1), metadata can help with all transformation steps:

- To provide information about schemata to which a set of structured data adheres or the syntax description needed to understand the signs transmitted.
- To advance the interpretation of data by providing context information (e.g. geographical or temporal).
- To point out related (web-) resources to broaden the context of an information (e.g. links to similar datasets or related web site).

2.1.5 The FAIR Data Principles

In 2014, a workshop in Leiden, Netherlands, was held, named "Jointly Designing a Data Fairport". A wide group of academics and representatives of companies and other organisations concluded the workshop by drafting a concise and measureable set of principles to overcome common obstacles, impeding data discovery and reuse of (scientific) data.

The **FAIR** Guiding Principles [7]

F To be Findable:

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

2 data are described with rich metadata (defined by R1 below)

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

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

A To be Accessible:

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

i. the protocol is open, free, and universally implementable

ii. the protocol allows for an authentication and authorization procedure, where necessary

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

I To be Interoperable:

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

2 (meta)data use vocabularies that follow FAIR principles

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

R To be Reusable:

1 (meta)data are richly described with a plurality of relevant attributes

2 (meta)data are released with a clear and accessible data usage license

3 (meta)data are associated with detailed provenance

4 (meta)data meet domain-relevant community standards

2.2 Semantic Web

I presented the common understanding of how data can herald information and knowledge in section 2.1.1. I refrained from specifying which step or state might be restricted to humans or machines. None of those restrictions would probably be correct. While I don't want to elaborate on the question of: "Can machines have Knowledge?", I do state that machines can glean information from data. To interpret a message and derive meaning is not limited to the human mind. All what is needed is context and an understanding of the concepts a domain is constituted of (semantics).

In 2001, Tim Berners-Lee, Hendler, and Lassila layed out their expectation of how the World Wide Web will eventually extend to become a Semantic Web [19]. The simple extension of web resources with structured, well defined data (or metadata) would give meaning to resources, previously only decipherable by humans. To identify these data resources uniquely by Universal Resource Identifiers (URI) and provide links to other resources, would be the first steps in the direction towards a "web of data that can be processed directly and indirectly by machines". quote?

> "The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning. better enabling computers and people to work in cooperation." [19]

Tim Berners-Lee is the director of the World Wide Web Consortium[24] (W3C), which is responsiblefor the development of standards for the World Wide Web, rewrite and by extension for the Semantic Web. At its core, the Semantic Web is defined by a collection of standards, which are *Recommendations* by the W3C. "Semantic Web technologies enable people to create data stores on the Web, build vocabularies, and write rules for handling data." [25].

2.2.1 Resource Description Framework (RDF)

This foundational technology of the Semantic Web and recommendation of the W3C [20] is used to describe any resource, using URIs as identifiers. Resources are defined by set of characteristics which are expressed as attributes or relations to other resources. Statements in the form of "subject, predicate, object" (named "triples") are used to convey such characteristics. The resource described is uniquely identified by the URI of the subject. The object corresponds to the

[24] https://w3.org
[25] https://w3.org/standards/semanticweb/

content or reference of the statement. Literals (strings) are used to serialize content (or values), URIs of resources provide the target of a reference. The predicate is the semantic link between the subject and the object and defines the meaning of this statement. This simple linguistic construct makes RDF data understandable for humans and machines alike:

```
<http://dbpedia.org> publisher "DBpedia Association".
```

This triple describes the resource http://dbpedia.org (identified by the URI of the subject). Its object is the literal "DBpedia Association" connected to the subject with the predicate: publisher. Without the predicate, no meaning could be ascribed to the datum "DBpedia Association", as the type of relation between subject and object would be unknown. Thus, the predicate is what lends meaning to a statement.

The same information about the publisher of a website could be expressed differently in RDF data model. Since "DBpedia Association" represents an organisation, it could be introduced and described as an instance of a concept named "Organisation". This instance is capable of providing multiple attributes, not only its name:

```
<http://dbpedia.org> publisher <http://dbpedia.org/DBpediaAssociation>.
<http://dbpedia.org/DBpediaAssociation> type "Organisation".
<http://dbpedia.org/DBpediaAssociation> name "DBpedia Association".
<http://dbpedia.org/DBpediaAssociation> headquarter "Leipzig".
```

It is obvious that the instantiation of the objects "Organisation" and "Leipzig" would bear the same benefits. By extending this list of statements, describing additional resources and their characteristics, a directed graph is constructed. The data of this graph is highly interlinked and has the ability to relate to resource descriptions outside this graph as well. A URI identifies a resource in the world without ambiguity (when carefully constructed based on existing domain names). This allows linking of data objects without regard to resource locations (datasets, service endpoints etc.) and institutions, creating, what is called, "Linked Data" (section 2.2.3).

While labels like "publisher" have meaning for humans, they are ambiguous and could be misinterpreted. Especially machines cannot resolve this ambiguity and could not infer meaning from such statements. To address this issue, predicates are also identified by URIs that can be looked up for further information. Predicates are called properties in the RDF data model.

```
<http://dbpedia.org> <http://purl.org/dc/terms/publisher> <http://dbpedia.org/DBpediaAssociation>.
<http://dbpedia.org/DBpediaAssociation> <http://purl.org/dc/terms/name> "DBpedia Association".
```

Sets of properties can be defined and documented by institutions, like DCMI (section 2.1.4). They can then be reused by others, increasing interoperability

and reusability. These sets of properties together with associated concepts and annotations are called ontologies.

2.2.2 Web Ontology Language (OWL)

The Web Ontology Language is a W3C recommended standard [21] based on the RDF data model, which could be summarized as an ontology for defining ontologies. An ontology is a set of concepts, properties and logical axioms, with which to model a domain of knowledge or discourse. This conceptualization of a given domain or idea, allows for a formal representation with RDF as well as its automatic interpretation by machines.

Concepts are classes under which all objects of a domain can be classified, dividing up the domain in abstract objects. Properties provide meaning for the links to instances of concepts or literal objects (values), which, in turn, lends meaning to the concepts themselves. Additionally, subclass relations and restrictions on properties (e.g. domain and range definitions) help to specify more complex relations of a domain. This semantic layer between the domain knowledge on one side and its representation in RDF on the other is free of restrictions imposed by underlying technologies, distinguishing it from other data models (e.g. database schemata). A wide range of ontologies are available for any type of domain. Upper ontologies are general use vocabularies (such as Dublin Core section 2.1.4) which can be reused together with other ontologies. High reusability is another difference to many other domain description languages.

Multiple langauge profiles are available [22], introducing different logical regimes to OWL, such as Description Logic (OWL-DL) or based on Rule Languages (OWL2-RL). Profiles allow for reasoning over RDF data, adhering to ontologies under such regimes. The profile OWL Full is an extension of the RDF Schema (RDFS [23]), which provides basic elements for the description of ontologies (allowing for class hierarchies and basic relations).

Description Logic is a fragment of first-order predicate logic [24] and a formalism for representing knowledge. OWL is heavily influenced by Description Logic (projects like DAML+OIL [25]) in order to achieve a beneficial trade-off between language expressiveness and computational complexity of reasoning.

2.2.3 Linked Data

Linked Data is the idea of how data is freed from an islands of unconnected data, so it can be used and referenced all over the world referencing simply

their URIs. Links between data objects from different sources are not bound to restrictions, authorisation procedures, licensing or any technical obstacles, they simply state that: "There is a data object (published in a well defined manner - e.g. RDF) which is related and it is identifiable on the Web with this URI". Machines and humans can follow up these links and explore the "Web of Data", expanding the contextual information.

needs link

> "Technically, Linked Data refers to data published on the Web in such a way that it is machine-readable, its meaning is explicitly defined, it is linked to other external data sets, and can in turn be linked to from external data sets." [26]

Additionally, a set of best practices were contrived by Tim Berners-Lee to identify the necessary steps needed to publish and link structured data on the Web, since "a surprising amount of data isn't linked in 2006, because of problems with one or more of the steps" [27].

1. Use URIs as names for things.
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL).
4. Include links to other URIs. so that they can discover more things.

Linked Data extends the demands of RDF as a data model by specifying HTTP as the access layer of choice and requiring the openness of the resource to be regarded of high quality. The result is a Web of Data, a machine-readable, semantic network of structured data, opposed to the HTML based web of documents (from humans, for humans).

quote

3 Related Work

3.1 Dataset vocabularies

This section is dedicated to dataset metadata vocabularies and application profiles [26], to compare them and list their (dis-) advantages.

Based on the FAIR Data Principles (section 2.1.5) and the list of important aspects of dataset metadata (section 1.1), I contrived the following list of aspects, against which I want to evaluate each vocabulary.

A **The vocabulary encourages the use of richly described and machine-readable resources.** Concepts are defined as exhaustive as necessary to describe all relevant aspects, avoiding free text properties in general. For example, replacing a literal with a well structured instance of `foaf:Agent`.

B **The vocabulary assigns globally unique URIs to metadata resources.** Demanding URIs as identifiers, independent of the chosen data representation (even for non RDF or XML metadata).

C **The vocabulary can describe data access related properties and restrictions, enabling access for humans and machines alike.** Sufficient effort has been made to describe this important aspect in detail, considering all possible formats of a dataset.

D **The vocabulary can portray provenance information extensively.** Dataset provenance can be described extensively, including other datasets (e.g. sources), activities (e.g. data generation activities) and agents (e.g. publisher) as well as inter-relational properties between these concepts.

E **The vocabulary provides for detailed descriptions of rights and licenses.** Machine readable licenses are of utmost importance.

F **The vocabulary provides properties to cite identifiers of the data described.** The possibility to reference the data described directly (by identifier) in the metadata is available.

[26] a set of metadata elements defined for a particular application or other limited purpose, often based on a broader schema (like an ontology)

G **The vocabulary provides for qualified references between resources.** Relations between instances of dataset metadata can be qualified by roles (specifying the type of relations), time and other restrictions.

H **The vocabulary is easy to extend, to fit any given use case.** The vocabulary is general enough to fit any use case, it can easily be extended and no unnecessary restrictions, like restrictive cardinalities, are in place.

I **The vocabulary is unambiguous and easy to map to other metadata vocabularies.** The vocabulary is general enough to be able to match other metadata formats. Properties are defined clearly without overlapping the purpose of others (so users know which property to use).

J **The vocabulary offers additional properties to aid dissemination and discovery.** Extra properties are in place to provide keywords, genres, taxonomy concepts and general statements explaining the use of a dataset.

I will assign one of the following ratings to every item: **(2)** The requirement is supported in full. **(1)** The requirement is partially met. **(0)** The vocabulary does not support this requirement. While this list is helpful for evaluation and comparison purposes, the quality of a dataset vocabulary is also dependent on the intended domain of use and other factors, which I will mention as well.

3.1.1 The Data Catalog Vocabulary (DCAT)

In [28] the authors introduce a standardised interchange format for machine-readable representations of government data catalogues. The Data Catalog Vocabulary (DCAT) is a W3C recommendation [1] and serves as a foundation for many available dataset vocabularies and application profiles. Vocabulary terms for DCAT are inferred from the survey on seven data catalogues from Europe, US, New Zealand and Australia.

> "By using DCAT to describe datasets in data catalogs, publishers increase discoverability and enable applications easily to consume metadata from multiple catalogs. It further enables decentralized publishing of catalogs and facilitates federated dataset search across sites." [1]

DCAT defines three levels of abstraction, based on the following distinctions: A dataset describes a "collection of data, published or curated by a single agent, and available for access or download in one or more formats"[1], and represents the commonalities and varieties of the data held within (the 'idea' or intellectual content of that dataset). A Dataset is part of a data catalogue, representing

multiple datasets (e.g. of an organisation). Datasets manifest themselves (are available) in different forms (such as files, service endpoints, feeds etc.), expressed with the class `dcat:Distribution`. A dataset might be available for download at two different locations on the Web and available to be queried through an API endpoint. This scenario can be described by using three different `dcat:Distribution` instances.

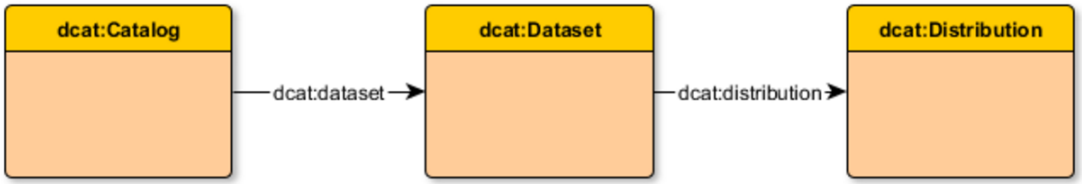


Figure 3.1: The basic idea of DCAT

This basic idea of differentiating between catalogue, dataset and distribution, has prevailed throughout the metadata domain for digital resources (not only datasets), and has become an quasi-requirement for metadata representations of web resources[27]. In fact, the general approach the authors of DCAT took, makes it possible to describe any digital object. This is a highly desirable feature, supporting **`extensibility`** and **`interoperability`** of the vocabulary.

As stated before (section 1.1), a downside to this approach is the possible unspecificity of resources, especially in regard to machine-readability, where uncertainty about formats is problematic.

Insufficient provenance information:

- DCAT expresses provenance in a limited way using a few basic properties such as `dct:source` or `dct:creator`, which can not be further qualified.
- No possibility to specify activities involved in the creation of datasets.
- There is no support or incentive to describe source datasets, related publications or conversion activities of transformations responsible for the dataset. This lack is crucial, especially in a scientific contexts, as it omits the processes necessary to replicate a specific dataset.
- Insufficient portrayal of context information (e.g. licenses, geography etc.)

Missing relations between Datasets:

- in general: Referencing related datasets is only possible on a very generic scale (e.g. `dct:relation`).

[27] `https://www.w3.org/TR/dwbp/#context`

- hierarchical: No inherent portrayal of dataset hierarchies is possible.
- evolutionary: No versioning pointers between dataset representations.

Relations to agents are too cursory:

- A very restricted number of properties pointing out agents, without any further qualification (e.g. `dct:publisher`, `dcat:contactPoint`), other related entities, like software, projects, funding etc., are neglected all together.
- No agent role concept to define new relations.

Technical description of distributions is lacking, restricting the accessibility of the data:

- Only superficial attributes for describing the technical characteristics of a distribution are available.
- No access information are available, such as access restrictions or service description, needed to describe service endpoints.
- No specificity when describing serialisation or media type of a distribution (e.g. file format).

General lack of specificity, inviting non-machine-readable expressions of resources:

- Insufficient specificity of property ranges (e.g.: `dct:license`, `dct:temporal`, `dct:spatial`, `dct:language`, `dcat:mediaType`), thereby neglecting exactness of relevant metadata resources, such as licenses.
- A lack of referential and functional integrity due to missing role-based qualifications of properties (such as `dct:maintainer`) [29].

While this seems to be an extensive list of shortcomings, most of the points listed above are due to the general approach of DCAT. The list merely indicates that portraying dataset metadata with DCAT alone, might not be sufficient for most domains and use cases.

In turn, extending DCAT as upper ontology provides a sufficient basis for any metadata descriptions, without regard to domain or use case. Adopting not only its basic ideas, but the aspects of easy **`extensibility`** and **`interoperability`**, should prove beneficial. Addressing the issues with **`provenance`**, **`licensing`** and **`access`** as well as domain specific demands for metadata is the central task I set out to complete.

The evaluation table is reused in the eval section. The numbers within have to be revisited...

Requirement	A	B	C	D	E	F	G	H	I	J	Sum
Evaluation of DCAT	2	2	0	0	0	1	0	2	2	2	11

3.1.2 Vocabulary of Interlinked Datasets (VoID)

The Vocabulary of Interlinked Datasets (VOID) [30] is widely accepted and used within the Semantic Web community, for instance in projects such as: OpenLink Virtuoso[28] , LODStats[29] , World Bank[30] and others. VOID can be used to express general metadata, access metadata, structural metadata and links between Linked Data datasets. Tools to create VOID metadata are described in [31] where authors also presents techniques of reduction in order to create descriptions for Web-scale datasets. In the same paper the importance of VOID is well established but there is still a lack of important metadata which is not described, for example license and provenance. For simple datasets VOID performs well, which is supported by the fact of wide acceptance of the vocabulary. However, in a case of complex datasets VOID is not expressive enough. In particular, access metadata includes the `void:dataDump` property, which points to the data files of the particular dataset. This property should link directly to dump files as described in W3C Interest Group Note: Describing Linked Datasets with the VOID Vocabulary[31]. Thus, additional semantic information about the data files and the structure of the dataset can not be expressed using VOID. Moreover, VOID does not provide a distribution concept depriving the vocabulary of an important level of abstraction. Yet, this ontology is useful especially for Linked Data datasets, offering many useful statistical properties (such as `void:triples`) and the very handy concept of `void:LinkSet` describing the particular relation between datasets, where one holds links to instances of the other.

Requirement	A	B	C	D	E	F	G	H	I	J	Sum
Evaluation of VOID	2	2	1	1	0	0	1	1	1	1	10

3.1.3 Comprehensive Kerbal Archive Network (CKAN)

Metadata models vary and most of them do not offer enough granularity to sufficiently describe complex datasets in a semantically rich way. For example, CKAN[32] (Comprehensive Knowledge Archive Network), a data management

[28] http://virtuoso.openlinksw.com/
[29] http://stats.lod2.eu/
[30] http://worldbank.270a.info/
[31] http://www.w3.org/TR/void/
[32] http://ckan.org/

system used widely in (Open) Data Portals (such as datahub.io[33]) to provide web representations for datasets, operates on a JSON based schema[34] developed by the Open Knowledge Foundation[35]. CKAN allows simple access to a whole range of functions related to the management of datasets (such as search and faceting of data-sources) accessible via a REST interface. Its metadata schema has some similarities with DCAT: Datasets are collected under organisation objects, which are used as s primitive stand in for catalogues. Datasets have 'Resources' which assume a similar role as a distribution in the DCAT vocabulary.

Alas, there is no clear definition of the resource-object within CKAN documentation, nor are there any noteworthy restrictions. This has led to a medley of different use cases for this resource, where it assumes the role of a `dcat:Distribution` in one dataset (containing all data of the dataset) and providing different slices of the data in another example [32]. Furthermore, the extensive use of key-value pairs for additional data led to a shier host of (semi-) structured data, with only marginal agreement on key names between different Data Portals [32] adding to the general unclarity of this metadata format, complicating the mapping to other vocabularies.

This data model is semantically poor and inadequate for most applications consuming data automatically. I strongly discourage any organisation from adopting the CKAN format for their dataset metadata. Since it is used so frequently in Data Portals, I feel obliged to point out that there are mapping tools for most vocabularies to CKAN (as I provided for DataID in [8]).

Requirement	A	B	C	D	E	F	G	H	I	J	Sum
Evaluation of CKAN	2	1	0	0	0	0	0	0	1	1	5

3.1.4 Metashare

The META-SHARE ontology[33] is the offspring of a prior, XSD[36] based "metadata schema that allows aspects of [language resources] accounting for their whole lifecycle from their production to their usage to be described"[33]. META-SHARE differentiates between language resources (basically datasets with a language related purpose - text, audio etc.), technologies (e.g., tools, services) used for their processing and additional entities like reference documents, agents, projects or licenses. This allows for the portrayal of provenance in the

[33] `http://datahub.io/`

[34] `https://github.com/KSP-CKAN/CKAN/blob/master/CKAN.schema`

[35] `https://okfn.org`

[36] XML Schema Definition: a W3C recommendation for how to formally describe the elements in an XML document (https://www.w3.org/TR/xmlschema11-1/)

domain of Natural Language Processing. In addition it offers an exemplary way of describing licenses and terms of reuse[37]. Yet, META-SHARE is highly specialised for language resources, thus lacking generality and extensibility for other use cases. While not implementing the DCAT vocabulary, META-SHARE does provide an almost complete mapping to DCAT. Mappings to other ontologies might prove difficult, due to the large size of the vocabulary and the often employed (and ample) controlled vocabularies. The related META-SHARE XSD schema has been implemented in the META-SHARE web portal[38], providing many NLP related datasets for download.

Requirement	A	B	C	D	E	F	G	H	I	J	Sum
Evaluation of META-SHARE	2	2	1	1	2	0	0	0	1	1	10

3.1.5 Asset Description Metadata Schema (ADMS)

The Asset Description Metadata Schema[39] (ADMS) is a profile of DCAT, which is specialised to describe "Semantic Assets". Assets (as subclass of `dcat:Dataset`) are highly reusable metadata (e.g. code lists, XML schemata, taxonomies, vocabularies etc.) expressing the intellectual content of the data, which is represented (in most cases) in relatively small files.

ADMS adopts the DCAT structure and provides a well defined way of versioning between entities. Its specialised nature makes it unsuited for a broader approach of portraying datasets (as intended by the authors), but it can still contribute useful properties to DCAT based vocabularies (e.g. section 3.1.6). Since ADMS does not impose any restrictions it can be extended to DCAT without any consequences for DCAT based metadata documents. The evaluation below, therefore does not differ from DCAT.

Requirement	A	B	C	D	E	F	G	H	I	J	Sum
Evaluation of ADMS	2	2	0	0	0	1	0	2	2	2	11

[37] http://www.cosasbuenas.es/static/ms-rights/
[38] http://www.meta-share.org
[39] https://www.w3.org/TR/vocab-adms/

3.1.6 DCAT Application Profile for data portals in Europe (DCAT-AP)

The DCAT Application Profile for data portals in Europe[40] (**DCAT-AP**) is a specification based on **DCAT** (extended with **ADMS** properties) for describing public sector datasets in Europe. It was developed by a working group under the auspices of the European Commission. Its basic use case is "to enable cross-data portal search for datasets and to make public sector data better searchable across borders and sectors" [34]. This can be achieved by the exchange of descriptions of datasets among data portals.

Traits of the resulting profile[41] (version 1.1) released in October 2015 [35]:

- It proposes mandatory, recommended or optional classes and properties to be used for a particular application;
- It identifies requirements to control vocabularies for this application;
- It gathers other elements to be considered as priorities or requirements for an application such as conformance statement, agent roles or cardinalities.

DCAT-AP has been endorsed by the Standards Committee of ISA2[42] in January of 2016[43] for the use in data portals. Further, it has been implemented by over 15 open data portals in the European Union, including the European Data Portal[44].

In general, while some recommendation are in place (e.g. using **ODRL** license documents - section 3.2.2), **DCAT-AP** can not propose concrete improvements in extending **DCAT** to advance **`provenance`**, **`licensing`** or **`access`**. As remarked in section 7 of its specification[45], the representation of different agent roles is lacking in the current version of **DCAT-AP**. In my opinion, the second solution proposed within, using **PROV-O** (Provenance Ontology - see section 3.2.1), is the most comprehensive way of resolving this issue. Due to some cardinality restrictions (e.g. those on `dcat:accessURL`) and its specialisation for data portals, extending **DCAT-AP** to serve more elaborate purposes, can pose challenges.

Requirement	A	B	C	D	E	F	G	H	I	J	Sum
Evaluation of **DCAT-AP**	2	2	1	0	0	1	0	1	2	1	10

[40] https://joinup.ec.europa.eu/asset/dcat_application_profile/home
[41] https://joinup.ec.europa.eu/catalogue/distribution/dcat-ap-version-11
[42] https://ec.europa.eu/isa/isa2/index_en.htm
[43] https://joinup.ec.europa.eu/community/semic/news/dcat-ap-v11-endorsed-isa-committee
[44] https://www.europeandataportal.eu/
[45] https://joinup.ec.europa.eu/catalogue/distribution/dcat-ap-version-11

3.1.7 The HCLS Community Profile

The W3C interest group Semantic Web for Health Care and Life Sciences (HCLS) represents many stakeholders of the Life Sciences, seeking to "develop, advocate for, and support the use of Semantic Web technologies across health care, life sciences, clinical research and translational medicine" [36].

Their community profile (an ongoing effort by this W3C interest group[46]), extends DCAT with versioning and detailed summary statistics, through a three component model. This model introduces an additional abstraction level between dataset and distribution, the so called 'Version Level Description', which contains version specific properties (e.g. `dct:isVersionOf`). The profile is structured in multiple modules, dealing with different levels of specificity [37]:

Core Metadata captures generic metadata about the dataset, e.g., its title, description, and publisher.

Identifiers describes the patterns used for identifiers within the dataset and for the URI namespaces for RDF datasets.

Provenance and Change describes the version of the dataset and its relationship with other versions of the same dataset and related datasets, e.g., an external dataset that is used as a source of information.

Availability/Distributions provides details of the distribution files, including their formats, in which the dataset is made available for reuse.

Statistics used to summarise the content of the dataset.

The HCLS profile reuses 18 vocabularies with 61 properties [37], covering many goals of my evaluation. The chosen approach of this profile is sound and achieves qualitatively good metadata, with an emphasis on FAIR Data principles.

One problem is the large number of reused vocabularies with overlapping purposes, which cause difficulties when mapping to other vocabularies. Its cardinality restrictions can pose problems when extending the profile to use cases, especially outside the health care domain. Although, efforts were made to cover provenance in general, problems like qualifying otherwise static relations to agents or datasets can not be solved by the incorporated vocabularies. The approach to portray licenses does not improve DCAT.

A specific problem is the use of the property `dcat:accessURL`, which, according to the profile, could be used on the dataset abstraction level ('Summary Level Description'). This clearly violates the specification of DCAT. In general the

[46] `https://www.w3.org/blog/hcls/`

boarder between `dcat:Dataset` and `dcat:Distribution` has to be defined more carefully when adding an additional layer in between.

Requirement	A	B	C	D	E	F	G	H	I	J	Sum
Evaluation of HCLS Profile	2	2	1	1	0	2	0	1	1	1	11

3.1.8 CERIF

The metadata format CERIF[47] (Common European Research Information Format) is a metadata development, which began in the early 1990s. The shortcomings of its first approach were addressed by CERIF2000, which became a EU recommendation for its member states for research data. Since 2002 CERIF is further developed by the European Current Research Information Systems[48].

CERIF provides generalized base concepts and relations between them, based on an entity-relationship[49] approach [38]. Relations (or 'linking entities') are qualified with roles, temporal and spacial statements, supplemented with provenance and versioning information. In contrast to Dublin Core based metadata standards (i.e. DCAT), CERIF was developed with a particular focus on referential and functional integrity of resources, avoiding ambiguity in interpretation [29]. CERIF provides much more than just metadata for datasets, it addresses metadata needs for the whole science community, describing projects, funding, facilities, organisations and other.

This paper lines out the main characteristics of CERIF metadata [29]:,

- it separates clearly base entities from relationships between them and thus represents the more flexible fully-connected graph rather than a hierarchy;
- it has generalised base entities with instances specialised by role (e.g. <person> rather than <author>), in the linking entities;
- it handles multilinguality by design and temporal information (so representing versions) to the appropriate attribute treated as an entity (example <title> linked to <publication>);
- temporal information is in the link entities not the base entities (e.g. employment between two dates is in the linking relation between <person> and <organisation> and not an attribute of either of the base entities);

[47] `http://www.eurocris.org/cerif/main-features-cerif`

[48] `http://www.eurocris.org`

[49] Entity–relationship model: describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types.

- the temporal information in linking entities provides provenance and versioning recording (e.g. versions of datasets and – in the associated role attribute – the method of update or change);
- CERIF separates the semantics into a special 'layer' which is referenced from CERIF instances. The sematic layer includes permissible values for roles in any linking entity and for controlled values of attributes in base entities (e.g. ISO country codes). Thus semantic terms are stored once and referenced many times (preserving integrity).

CERIF offers a comprehensive approach for solving a host of metadata demands in a sophisticated manner. The chosen abstraction levels (layers) are appropriate and the adherence to an entity-relationship approach is arguably a working solution for qualifying relations. The main problem with CERIF is the complexity of its ontology[50] [51] together with the unique approach to metadata (unlike the DCAT based understanding of metadata). The imposes hurdles when studying, mapping and extending this ontology. Furthermore, the ontology proves to be not specific enough when dealing with information on access and licenses.

Requirement	A	B	C	D	E	F	G	H	I	J	Sum
Evaluation of CERIF	2	2	0	2	0	2	2	1	1	1	13

3.1.9 DataID version 1.0.0

The common shortcomings of dataset vocabularies revealed in this section were also afflicting the previous version (1.0.0) of the **DataID** ontology [8]. Rooted in the Linked Data world, it neglected important information or provided properties (e.g. `dataid:graphName`) which are orphans outside this domain.

While it already imported the important Provenance Ontology (PROV-O-section 3.2.1), to cover the general issues with **provenance**, it was lacking in regard to specificity of **access** and **licensing**. The narrow definition of datasets (i.e. restricted to Linked Data datasets) was inadequate for use cases outside this domain and so inhibited **extensibility**.

Requirement	A	B	C	D	E	F	G	H	I	J	Sum
Evaluation of DataID 1.0.0	2	2	0	2	1	0	0	1	2	1	11

[50] http://eurocris.org/ontologies/cerif/1.3/
[51] http://eurocris.org/ontologies/semcerif/1.3/

3.2 Secondary Literature

This section proffers a collection of associated literature, not directly touching on the subject of dataset metadata. Many subjects, such as representation of licenses and data quality, are relevant for providing metadata of datasets.

3.2.1 The Provenance Ontology (PROV-O)

> "Provenance is defined as a record that describes the people, institutions, entities, and activities involved in producing, influencing, or delivering a piece of data or a thing. In particular, the provenance of information is crucial in deciding whether information is to be trusted, how it should be integrated with other diverse information sources, and how to give credit to its originators when reusing it. In an open and inclusive environment such as the Web, where users find information that is often contradictory or questionable, provenance can help those users to make trust judgements." [39]

The Provenance Ontology[4] (PROV-O) is a widely adopted W3C recommended standard and serves as a lightweight way to express the provenance and interactions between activities, agents and entities (e.g. datasets).

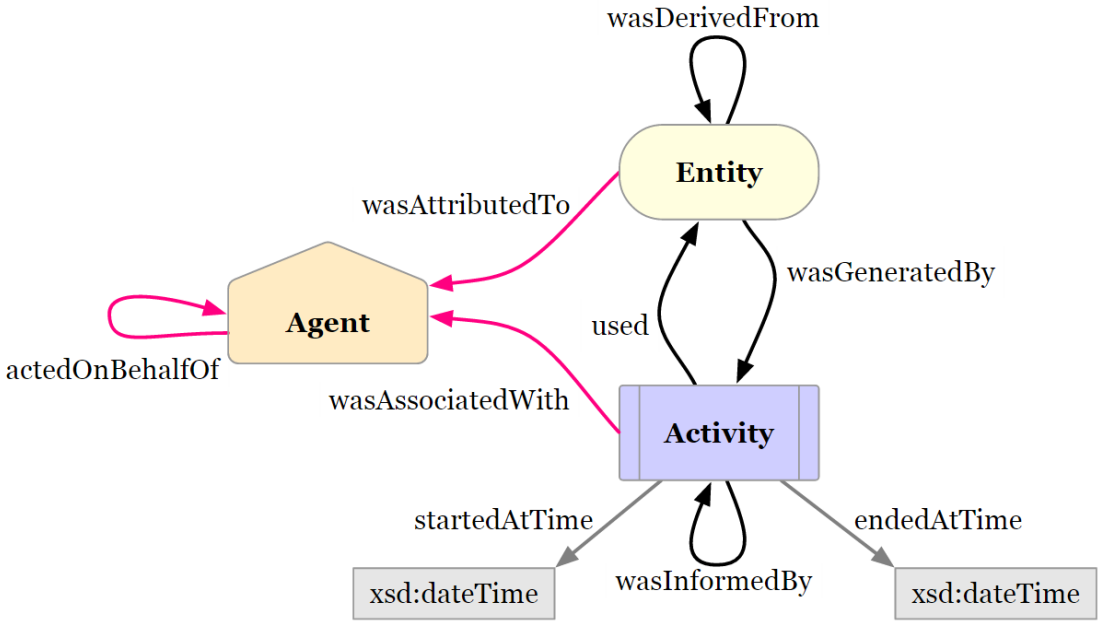


Figure 3.2: Linchpin of the Provenance Ontology: Entities, Agents, Activities [4]

In the context of datasets, provenance information on the activities which helped to create a dataset, which agents were involved in these processes and who to contact about it as well as source datasets or other entities involved are of interest, especially when trying to determain the trustworthiness of data. This ontology is "[...] the foundation to implement provenance applications in different domains that can represent, exchange, and integrate provenance information generated in different systems and under different contexts." [4]

Each of the relations depicted (fig. 3.2) have suitable qualification classes (e.g. `prov:Association`), so that a property (`prov:wasAssociatedWith`) can be inferred from the qualified property path (`prov:qualifiedAssociation/prov:agent`). Qualification classes provide qualification via properties such as `prov:role`.

In this example (from the official specification of PROV-O [4]), an association provides additional description about the :illustrationActivity that an agent named 'Derek' influenced:

```
@prefix prov:                  <http://www.w3.org/ns/prov#> .
@prefix :                      <http://example.org#> .

:illustrationActivity
   a prov:Activity;                  ## this illustration activity was
   prov:wasAssociatedWith :derek;    ##   associated with Derek in some way.
.

:derek a prov:Agent .

:illustrationActivity
   prov:qualifiedAssociation [       ## Qualify how the :illustrationActivity
      a prov:Association;            ##   was associated with the Agent Derek.
      prov:agent   :derek

      prov:hadRole :illustrationist; ## Qualification: The role that Derek served
      prov:hadPlan :tutorial_blog;   ## Qualification: The plan (or instructions)
                                     ##   that Derek followed when creating
   ].                                ##   the graphical chart.

:tutorial_blog   a prov:Plan, prov:Entity .
:illustrationist a prov:Role .
```

PROV-O will have a central role in the creation of DataID. Further reading on the key requirements, guiding principles, and design decisions which influenced the PROV Family of Documents[52] is advised (see [40]).

[52] https://www.w3.org/TR/2012/WD-prov-overview-20121211/

3.2.2 Open Digital Rights Language (ODRL)

The Open Digital Rights Language (ODRL)[53] is an initiative of the W3C community group with the same name[54], aiming to develop an open standard for policy expressions. The ODRL version 2.0 core model defines licensing policies in regard to their permissions granted, duties and constraints associated with these permissions as well as involved legal parties. Thus, an ODRL description allows to specify, in a machine-readable way, if data can be edited, integrated or redistributed.

3.2.3 Lexvo.org

Lexvo.org[55] is a service publishing language related information as Linked Data on the Web. The data published is conform to the Lexvo Ontology, providing unique identifiers for human languages in the context of geography, language families, words and word senses, scripts and characters [41]. All in all, the Lexvo dataset consists of over 8000 languages with a broad spectrum of language-related information that is extensively used by many data publishers and communities.

3.2.4 Friend of a Friend vocabulary (FOAF)

add foaf!

[42]

3.2.5 DataCite Ontology

DataCite is a "global non-profit organisation that provides persistent identifiers (DOIs) for research data [with the goal] to help the research community locate, identify, and cite research data with confidence"[56].

revise citation

DataCite published an XML schema for describing and citing research data[57], which is elaborate and at times novel approach for providing dataset metadata. Yet, due to its rigid XML structure with many cardinality restrictions, it does not feature in my collection of dataset metadata in this chapter.

[53] https://www.w3.org/ns/odrl/2/ODRL21
[54] https://www.w3.org/community/odrl/
[55] http://www.lexvo.org
[56] https://www.datacite.org/mission.html
[57] http://schema.datacite.org/meta/kernel-4.0/

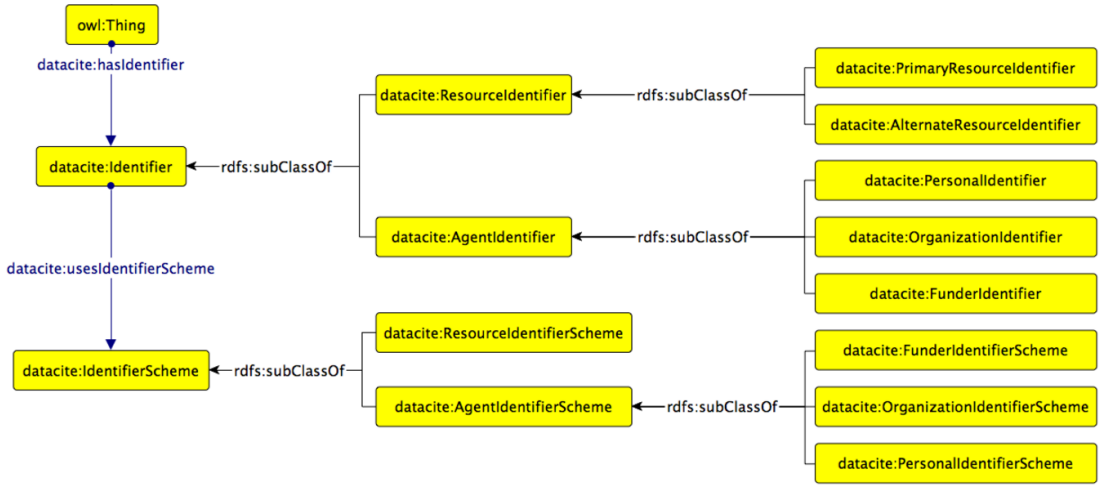


Figure 3.3: DataCite Identifier and IdentifierScheme

Of more interest, is the DataCite ontology which was published as an OWL ontology[58] and has a particular focus on representing identifiers (fig. 3.3). The Identifier class is divided into ResourceIdentifier and AgentIdentifier. In addition, an IdentifierScheme defines the format of the literal which represents the identifier. As opposed to an approach with data-types, this ontology allows for adding new schemes without altering the vocabulary itself and adding additional qualifications to the scheme entity [43]. Furthermore, the DataCite ontology contains a multitude of predefined IdentifierScheme instances, ready to be used.

3.2.6 re3data.org

The re3data[59] registry currently lists over 1.600 research repositories, making it the largest and most comprehensive registry of data repositories available on the web. By providing a detailed metadata description of repositories, the registry helps researchers, funding bodies, publishers and research organisations to find an appropriate data repository for different purposes[**PAMPLE2013**]. Initiated by multiple German research organisations, funded by the German Research Foundation[60] from 2012 until 2015, re3data is now a service of DataCite[61]. In 2014 re3data merged with the DataBib registry for research data repositories into one service[62]. The re3data project was initiated by the Library and Information

[58] http://www.sparontologies.net/ontologies/datacite/source.html
[59] http://www.re3data.org/
[60] http://www.dfg.de/
[61] https://www.datacite.org/
[62] http://www.re3data.org/tag/databib/

Services section (LIS) of the German Research Centre for Geosciences (GFZ), the library of the Karlsruhe Institute of Technology (KIT) and the Berlin School of Library and Information Science (BSLIS) at Humboldt-Universität zu Berlin.

One central goal of re3data is to enhance the visibility of existing research data repositories and to enable all those who are interested in finding a repository to assess a respective information service. This is achieved by an extensive and quality approved metadata description of the listed research data repositories. The basis for this description is the "Metadata Schema for the Description of Research Data Repositories", having 42 properties in the current version 3.0 [**r3dSchema**].

3.2.7 DBpedia

DBpedia is a community effort to extract structured information from Wikipedia and to make this information available on the Web, where it is used as a common point of reference for interlinking and enriching most of the structured data on the web today, establishing it as the center of the so called 'LOD cloud'.

this section needs polishing and references!

The main focus of the DBpedia extraction lies in mapping of info boxes, templates and other easily identified structured data found in Wikipedia articles to properties of an ontology. The DBpedia ontology is reflecting the relations and classifications found within Wikipedia. Together with the mappings to the Wikipedia XML templates, this ontology is curated by a community of interested people around the world to update schematical changes in time for the bi-yearly DBpedia releases.

Each release consists of around one hundred datasets for each of its 130 (2016) language editions (reflecting most of the languages of Wikipedia). Datasets are often created around particular properties (such as `rdf:type`) or other distinguishing features. To reflect this large corpus of structured data, the need for specific demands on its metadata are self-evident.

probably gets another field on the data quality vocabulary...

4 The DataID Ecosystem

4.1 Problem Statement

The inadequacies of current metadata vocabularies are manifold and diverse (section 3.1). As already introduced (section 1.1), there are some issues which protrude from the rest, due to their ubiquitousness in use cases or their import on aspects like interoperability, trustworthiness and governance of data.

This list of important aspects of metadata reflects these issues and explains them in detail:

(A1) **`provenance`**: a crucial aspect of data, required to assess correctness and completeness of data conversion, as well as the basis for trustworthiness of the data source (no trust without provenance).

(A2) **`licensing`**: machine-readable licensing information provides the possibility to automatically publish, distribute and consume only data that explicitly allows these actions.

(A3) **`access`**: publishing and maintaining this kind of metadata together with the data itself serves as documentation benefiting the potential user of the data as well as the creator by making it discoverable and crawlable.

(A4) **`extensibility`**: extending a given core metadata model in an easy and reusable way, while leaving the original model uncompromised expands its application possibilities fitting many different use cases.

(A5) **`interoperability`**: the interoperability with other metadata models is a hallmark for a widely usable and reusable dataset metadata model.

When regarding aspects **(A4)** and **(A5)**, taking into account the intricate requirements of many use cases (such as in **??**), **`extensibility`** and **`interoperability`** seem contradictory when leaving the more general levels of a domain description. A vocabulary capable of interacting with other metadata vocabularies might be too general to fit certain scenarios of use. Restrictive extensions to a vocabulary might encroach on its ability to translate into other useful metadata formats. This notion is corroborated by this document [44].

update ref

Note: I do not differentiate between **evolvability** and **extensibility** (as done in this source) in the context of this thesis. The discrepancies with **interoperability** are true for both concepts. Letting features 'die out' over time does not impact, in my understanding, the aspect of **extensibility**.

I conclude, not only is there a gap between existing dataset metadata vocabularies and requirements thereof, but it seems unlikely that one will be able to solve all these diverse problems with just one, monolithic ontology.

4.2 The multi-layer ontology of DataID

While trying to solve the different aspects, which I discussed in the previous section, and tending to the needs of different usage scenarios, the DataID ontology grew in size and complexity.

In order to keep the **DataID** ontology reasonable in size and complexity as well as not to jeopardise **extensibility** and **interoperability**, I modularised **DataID** in a core ontology and multiple extensions. The onion-like layer model (fig. 4.1) illustrates the import dependencies between the ontologies:

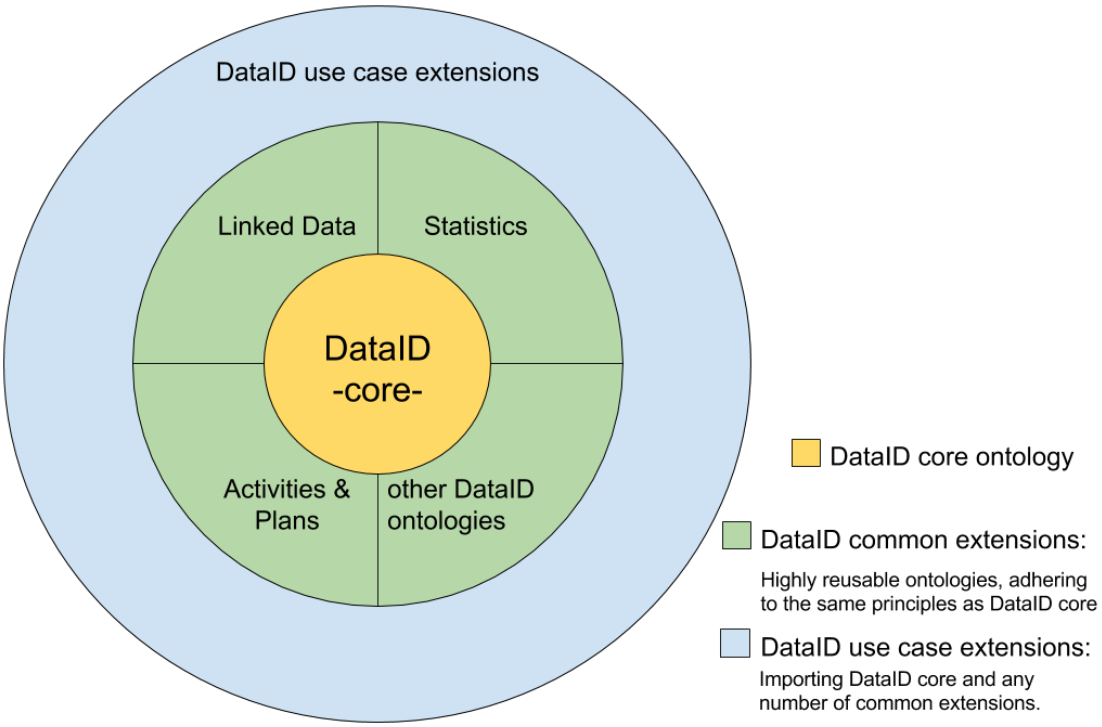


Figure 4.1: The Metadata Ecosystem of DataID

The scaling approach used to modularize the original **DataID** ontology adopts principles of the modular programming technique, separating concepts and properties of a large ontology into independent, interchangeable modules, specialised to fit common use cases, dependent only on **DataID core**. Thus, any vocabulary in this sphere must import **DataID core**, with the exception of **DataID core** itself. The mid-layer (or common extensions) of this model is comprised of highly reusable ontologies, extending **DataID core** to cover additional aspects of dataset metadata. Non of these are mandatory imports for any ontology, yet, in many cases some or all of them will be useful contributions. While I will not (and can not) impose any restrictions as to which ontologies not to import, ideally, ontologies of this layer should only import **DataID core**, to minimize discrepancies between mid-layer ontologies of different authors with overlapping purposes. The outermost layer of this sphere represents all vocabularies importing **DataID core** and any number if mid-layer ontologies and adding additional semantics to portray domain or use case specific demands for metadata.

The **DataID Ecosystem** is a suite of ontologies comprised of **DataID core** and its extensions, which were created to satisfy different use cases in a reusable manner. The brackets enclosed namespace prefixes are recommended and used throughout this work:

DataID core (dataid) provides the basic description of a dataset (cf. Section 5.1) and serves as foundation for all extensions to DataID.

Linked Data (dataid-ld) extends DataID core with an interface for the Sparql Service Description vocabulary [45], which not only allows to specify SPARQL endpoints, but offers properties useful for Linked Data datasets as well (such as `sd:defaultGraph`)[63]. While the VOID vocabulary is already imported in **DataID core** (see section 5.1), many VOID and properties only become useful in this context (e.g. `void:triples`). A specialised sub class of `dataid:Dataset` was introduced to better represent this type of data (`dataid-ld:LinkedDataDataset`).

Activities & Plans (dataid-acp) provides provenance information of activities which generated, changed or used datasets[64]. The goal is to record all activities needed to replicate a dataset as described by a DataID. Plans can describe which steps (activities, precautionary measures) are put in place to reach a certain goal. This extension relies heavily on the PROV-O ontology[4].

[63] https://github.com/dbpedia/DataId-Ontology/tree/master/ld

[64] https://github.com/dbpedia/DataId-Ontology/tree/DataManagementPlanExtension/acp

Statistics (dataid-md) will provide the necessary measures to publish multi-dimensional data, such as statistics about datasets, based on the Data Cube Vocabulary[46].

Other common extensions of similar general character as the ontologies of that layer, which could be useful in multiple use cases.

Ontologies under the DataID multilayer concept do not offer cardinality restrictions, making them easy to extend and adhere to OWL profiles (see also section 5.1). An application profile for a **`DataID`** service stack, which is beeing developed at the moment (chapter 10), was declared using SHACL [47].

Multiple requirements are planned to be enforced for the adoption of new ontologies in the common (or mid) layer of the DataID Ecosystem. They might contain (while not being restricted to):

- Authors must provide information about the reason for the new extension (and why the expected result is not achievable with existing extensions).
- Authors must document the Interoperability with other extensions of this layer (and where problems (such as semantic overlap) are to be expected).
- Authors must inform about conformity with OWL profiles.

Extending this ecosystem of dataset metadata with domain-specific OWL ontologies adds further opportunities for applications clustered around datasets. I will demonstrate the methodology of working with these ontologies to satisfy the metadata requirements of complex use cases in a best practice (section 6.2) as well as an application of these practices in chapter 7.

5 DataID core Ontology

5.1 Overview

This section will provide an overview of the **DataID core** ontology and some background on basic design decisions made. Most of the vocabularies reused by this ontology have already been presented in chapter 3 and need no further introduction.

DataID core developed out of a previous version of **DataID** by following the aspiration to modularise **DataID** into a core-ontology, surrounded by its dependents. Many design decisions of **DataID core** were already introduced in its former version, which was designed to make DCAT, combined with the VOID vocabulary, fit the requirements of the DBpedia use case for a hierarchy of Linked Data datasets (see also section 3.2.7):

> "The DBpedia dataset, with its different versions and languages, multiple SPARQL endpoints and thousands of dump files with various content serves as one example of the complexity metadata models need to be able to express. We argue that the DCAT vocabulary as well as the established VoID vocabulary only provide a basic interoperability layer to discover data. In their current state, they still have to be expanded to fully describe datasets as complex as DBpedia [...]." [8]

DataID core is founded on two pillars: the DCAT and PROV-O ontologies. To incorporate DCAT as the basis of **DataID**, to further extend DCAT with PROV-O, introducing extensive provenance records in the process, and adding properties specific to the DBpedia use case was the original premise of this endeavour. In addition, the VOID vocabulary was adopted to cover Linked Data specific semantics and provide more general properties, such as `void:subset` for establishing dataset hierarchies. The wide application of these vocabularies in the context of the Semantic Web was the rational behind these decisions, furthering the goal of **interoperability**.

As DCAT, **DataID core** is centred around the Dataset and Distribution concepts which were imported from DCAT. I introduced the class

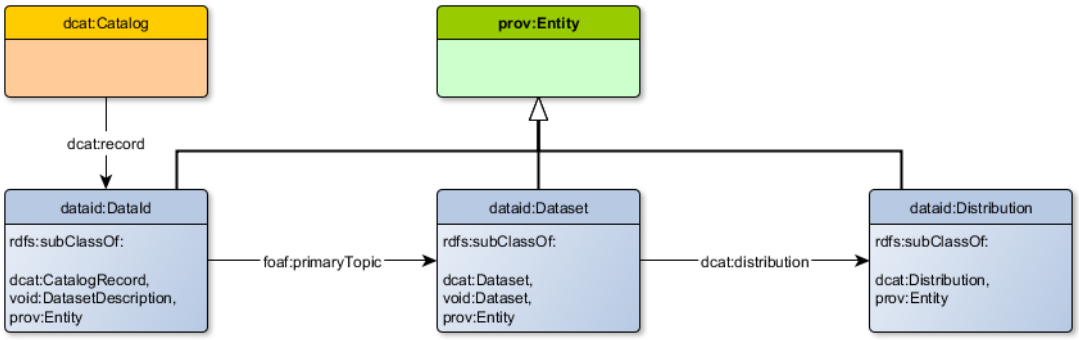


Figure 5.1: Foundations: Combining DCAT, VOID and PROV-O

`dataid:DataId` (see section 5.2.1), merging both `dcat:CatalogRecord` and `void:DatasetDescription` into one, and providing an additional level of abstraction between `dataid:Dataset` and `dcat:Catalog`. Instances of this class can be compared to root elements in the XML world, since all subsequent instances, describing a dataset, are hierarchicaly below a DataId instance. Though, this comparison is of course misleading, since a graph has no orientation. Yet, most of DataID documents[65] will have structural similarities to XML documents.

All dataset related classes of **`DataID core`** (`dataid:DataId`, `dataid:Dataset` and `dataid:Distribution`) are sub classes of `prov:Entity`, which is the interface needed to harness the possibilities of the Provenance Ontology (PROV-O). In the context of this description of **`DataID core`**, the word 'Entity' is used to refer to instances of `prov:Entity` (ergo: instances of these three classes).

With PROV-O, describing the circumstance of provenance for any domain is possible (section 3.2.1). Central to this, is the concept of qualification classes, providing qualifications for more general properties (e.g. for `prov:wasAttributedTo`). Describing the interrelations between Entities (such as a particular dataset or just a single distribution of it) and Agents (i.e. a person or an organisation) are salient requirements in most environments for datasets and their metadata. Thus, **`DataID core`** has singled out these relations to further qualify them and provide much needed referential integrity.

explain

For example, the property `dataid:associatedAgent` (which is a sub property of `prov:wasAttributedTo`) is a universal relation between an Entity and an Agent. It can (and should) be qualified by an instance of `dataid:Authorization`, a sub class of `prov:Attribution`. An Authorization adds qualifications and restrictions to the original property, such as an agent role (defining the role the agent

[65] DataID graph serialised as a metadata document

has in regard to the Entities involved - see section 5.2.6). core allows for assigning Actions to an Authorization, which specify what an Agent can do and for which tasks he/she/it is responsible for.

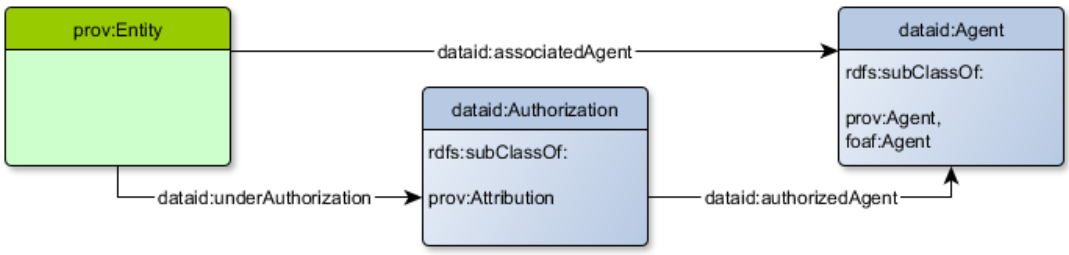


Figure 5.2: Foundations: Using PROV-O to qualify properties

In its current version 2.0.0, **DataID core** offers a general approach for describing dataset metadata, incorporating important ontologies to extend DCAT with **provenance**, qualifying relations and their intended range classes, hierarchical dataset structures, management of rights and responsibilities of agents and exhaustive descriptions for data **access**.

In general, **DataID core** waived the use of cardinality restrictions[66], making it easier to adhere to the OWL 2 RL[67] profile and maintaining the benefits of easy **extensibility** and **interoperability** prevalent with DCAT. In turn, **DataID core** restricts some of the very general property ranges of Dublin Core and DCAT properties (such as `dct:license` or `dcat:mediaType`), to reduce impreciseness and increase machine-readability.

DataID core is making use of the following namespace:

```
http://dataid.dbpedia.org/ns/core#
```

At this URI, an extended description of each class and property, introduced by this ontology, can be found as well as the complete ontology specification in Turtle[68] or OWL[69] serialization.

should i put the whole specification in the appendix? would be about 20 pages...

For a better understanding of the imported concepts and properties, which are reused by **DataID core**, I would advise to read up on DCAT [1] and PROV-O [4] (since these are used most commonly), to gain a more complete picture of the underlying structure and rational of these ontologies. While I gave a concise introduction to both in section 3.1.1 and section 3.2.1 respectively, this work is not the space to repeat this in more detail.

[66] this is the purpose of an application profile, not of an ontology
[67] https://www.w3.org/TR/owl2-profiles/#OWL_2_RL
[68] http://dataid.dbpedia.org/ns/core.ttl
[69] http://dataid.dbpedia.org/ns/core.owl

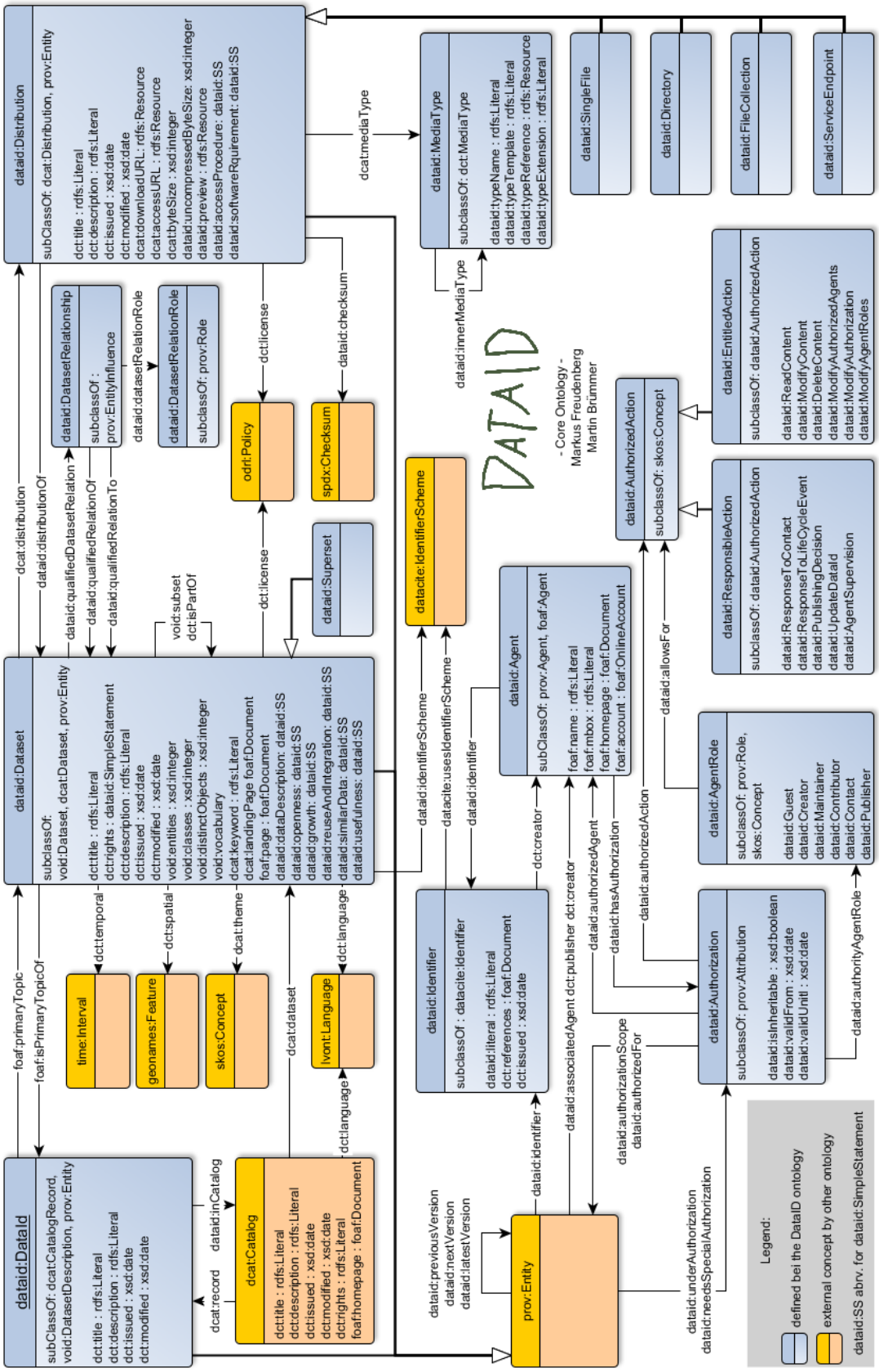


Figure 5.3: `DataID core`

5.2 Classes

This section is partitioned by concepts to introduce the **DataID core** ontology. Each class is presented with an item of written comment about the general reasoning behind its existence and its intended use. For illustration purposes, a running example was woven into the descriptions of concepts and properties. This example is a reduced version of an original **DataID** document of the Arabic DBpedia (release: 2015-10[70]). Under the main dataset, only two sub-datasets are shown (as opposed to over 50 in the real world example from which this is drawn). It was chosen to cover many aspects of **DataID core** and to provide an easy use case which could arise in a similar fashion outside the DBpedia domain. The full example is available in Turtle serialisation[71] in Appendix I. The basic structure of its **DataID** document is outlined below (Figure 5.14).

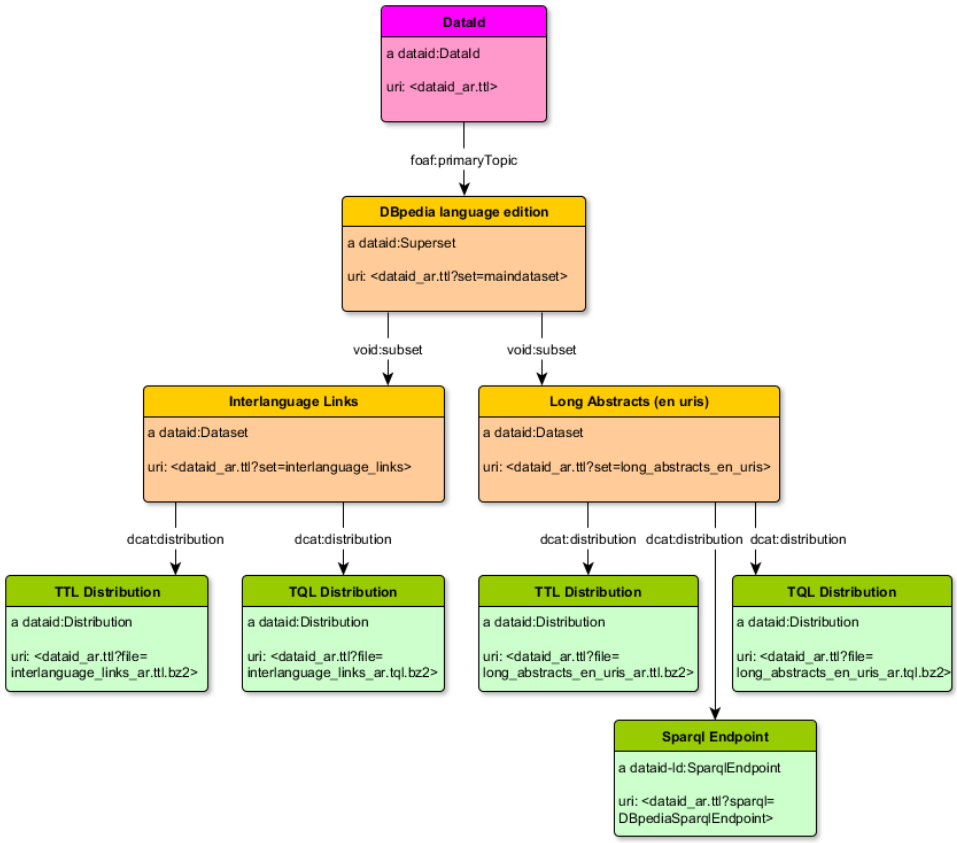


Figure 5.4: Schematic view: A shallow hierarchy of Datasets.

[70] http://
[71] http://

For the purpose of this running example the following base URI was used to shorten all subsequent URIs of this:

```
@base <http://downloads.dbpedia.org/2015-10/core-i18n/ar/2015-10_>
```

true?

Some instances referenced in the example were left out to reduce redundancy.The example omits the more common properties of Dublin Core[72] and RDFS[73] such as `dct:title`, `dct:description`, `dct:modified`, `dct:issued` and `rdfs:label` to make this example more easy to read.

In addition, this running example does not only use **`DataID core`**, it also introduces a first **`DataID`** extension from the common layer of the **`DataID Ecosystem`** (section 4.2) for Linked Data datasets. The Linked Data extension[74] (prefix: dataid-ld) is used to describe dataset attributes, specific to Linked Data.

The remainder of this section features each subsection with a short summary, stating the purpose of each concept, a list of properties commonly used to describe instances of this concept, a schematic depiction of the concept, followed by at least one example instance of this class (taken from the running example of DBpedia). The list of properties features a rating for each property to advise whether an application profile based on **`DataID core`** should declare a property to be: mandatory **(M)**, recommended **(R)** or optional **(O)**.

5.2.1 DataId

The class `dataid:DataId` inherits from `dcat:CatalogRecord`, which does not represent a dataset, but metadata about a dataset's entry in a catalogue. Additionally, in the context of **`DataID core`**, it represents metadata about a **`DataID`** document (graph), such as version pointers, modification dates and relations to its context (such as Agents, Catalogues, Repositories). This DataId resource is the most abstract Entity in any **`DataID`** graph.

Properties specifically used with `dataid:DataId`:

`dataid:inCatalog`: the inverse of `dcat:record` references the DCAT catalogue a **`DataID`** is entered in. **(R)**

`foaf:primaryTopic`: this functional property is used to point out the dataset resource this **`DataID`** represents. **(M)**

In addition, the following list contains properties which can be used with any Entity and are presented here for convenience:

[72] `http://`
[73] `http://`
[74] `https://github.com/dbpedia/DataId-Ontology/tree/master/ld`

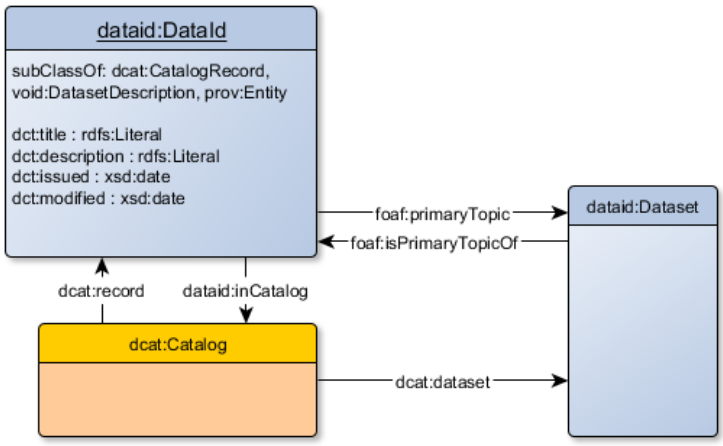


Figure 5.5: Class DataId

`dataid:associatedAgent`: points out all agents which have some (unspecified) influence (or authority) over this Entity. **(O)**

`dataid:underAuthorization`: refers an instance of `dataid:Authorization` which qualifies the generic `dataid:underAuthorization` relation (see section 5.2.6). **(R)**

`dct:publisher`: While the PROV-O based way to declare associated agents and their roles in regard to this Entity is in place, using established properties to point out agents (such as `dct:publisher`) is not redundant and should be kept as best practice. **(R)**

`dataid:nextVersion`: is used to identify instances of the same type (in this case `dataid:DataId`), which are next in the chain of different versions of this resource. **(O)**

`dataid:previousVersion`: points out the previous version of this instance. **(R)**

`dataid:latestVersion`: points out the latest version of this instance. **(R)**

The example below illustrates the use of this concept. It provides a pertaining dataset catalogue (via `dataid:inCatalog`), uses version pointers to define its position in a branch of a versioning system, points out agents and authorizations and the dataset about which this **`DataID`** is about (`foaf:primaryTopic`).

```
<dataid_ar.ttl>
        a                           dataid:DataId ;
        dataid:associatedAgent      <http://wiki.dbpedia.org/dbpedia-association> ;
        dataid:inCatalog            <http://downloads.dbpedia.org/2015-10/2015-10_dataid_catalog.ttl>
             ;
        dataid:latestVersion        <http://downloads.dbpedia.org/2016-04/core-i18n/ar/2016-04
             _dataid_ar.ttl> ;
        dataid:nextVersion          <http://downloads.dbpedia.org/2016-04/core-i18n/ar/2016-04
             _dataid_ar.ttl> ;
        dataid:previousVersion      <http://downloads.dbpedia.org/2015-04/core-i18n/ar/2015-04
             _dataid_ar.ttl> ;
        dataid:underAuthorization   <dataid_ar.ttl?auth=creatorAuthorization> ;
        dct:hasVersion               <dataid_ar.ttl?version=1.0.0> ;
        dct:publisher                <http://wiki.dbpedia.org/dbpedia-association> ;
        dct:title                    "DataID metadata for the Arabic DBpedia"@en ;
        foaf:primaryTopic           <dataid_ar.ttl?set=maindataset>
```

Listing 5.1: Instance of a DataId

5.2.2 Dataset

This is the central concept of the **`DataID core`** ontology and offers a multitude of useful properties to describe a dataset comprehensively.

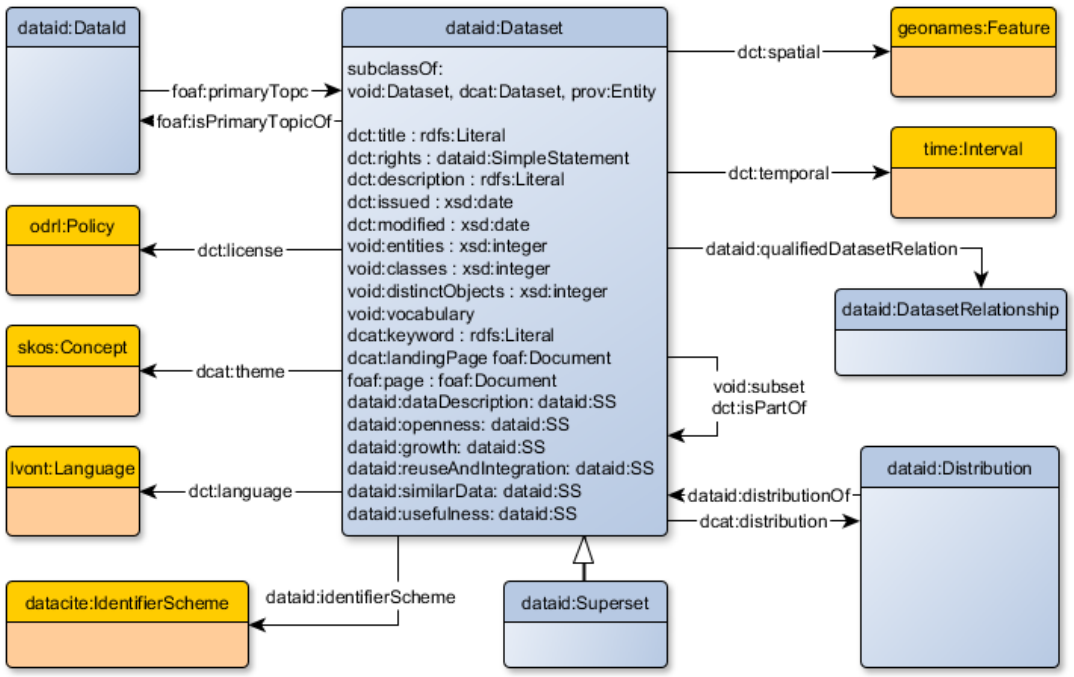


Figure 5.6: Class Dataset

The dataset concept of both the DCAT and VOID were merged into `dataid:Dataset`, providing useful properties about the content of a dataset from both ontologies. In particular, the property `void:subset` allows for the creation of dataset hierarchies, while `dcat:distribution` points out the distributions of a dataset. The `dataid:Superset` as a subclass of `dataid:Dataset` shall be used to represent multiple sub-dataset entities, portraying dataset collections or hierarchical dataset structures in general. Opposed to a conventional Dataset, a Superset is prohibited from possessing Distributions (referred to with `dcat:distribution`). It is strongly recommended that each Dataset shall either have at least one Distribution or one Sub-Dataset.

In the running example, the main dataset (an instance of `dataid:Superset`), is used as a hierarchical root, representing all Sub-Datasets clustered around a common topic. In the case of DBpedia, all Datasets were arranged under a Superset representing a DBpedia language edition.

Properties commonly used with instances of this class are:

`dcat:distribution` provides the distributions of a dataset and should be used distinct from `void:subset`. **(M for non Supersets)**

`void:subset` is the property used to reference sub datasets which are part of a superset. Its use is limited to the sub class `dataid:Superset`. **(M if Superset)**

void subset only for superste???

`dct:license` as already stated (**??**), this property is restricted in the context of **`DataID`** to instances of the class `odrl:Policy` of the ODRL ontology, providing machine-readable licensing information. **(M)**

`dct:language:` This property from the Dublin Core vocabulary is used to point out the predominant language used within a dataset. Since a `rdfs:range` statement is not provided, even literals could be used with this predicate. **`DataID core`** restricts the range of this property to instances of lvont:Language, a class from the Lexvo ontology (see section 3.2.3), describing human languages in a machine readable way. **(R)**

`void:vocabulary` in the VOID vocabulary, this property is used to point out the associated ontology used to create a dataset. **`DataID core`** broadens its use case so that any schema document (e.g. XSLT or database schemata) can be pointed out, depending on the type of data. **(R)**

`foaf:page` is commonly used to point out web pages which have the function of a manual or documentation for the resource at hand. **(R)**

dct:rights this property is used to provide a statement or resource about issues like copy rights or legal notes. To avoid unqualifiable literal statements, this property is reduced to `dataid:SimpleStatement`. **(O)**

foaf:isPrimaryTopicOf is the inverse property of `foaf:primaryTopic` (see section 5.2.1). **(O)**

dataid:relatedDataset: generic pointer to related datasets (sub property of `dct:relation`), qualifiable by `dataid:qualifiedDatasetRelation`. Refer to **??** for more. **(O)**

dataid:qualifiedDatasetRelation provides an instance of concept `dataid:DatasetRelationship` which is a qualification class for the generic property `dataid:relatedDataset` **(O)**

dataid:identifierScheme: provides a resource describing the identifiers used within the dataset to uniquely identify a record (data point, datum). Refer to section 5.2.9 to learn more about Identifiers. **(O)**

dcat:keyword: a simple keyword or tag associated with this dataset **(O)**

dcat:landingpage: references a web page of general character (such as the homepage of an organisation) **(O)**

Multiple properties for textual statements on different aspects of a Dataset were created. All of which provide publishers, maintainers and other agents a way to convey statements of general character on such subjects as described below. This information will be useful in many scenarios related to dissemination tasks, for example, those described by the Horizon 2020[75] data management plan guidelines[76] (more on this in chapter 7).

All of these properties are listed below have the common `rdfs:range` of `dataid:SimpleStatement`, which allows to either provide a literal or reference a web page containing the textual information about the resource (see section 5.2.10).

dataid:dataDescription: provides a detailed textual description of the data represented by this Dataset. **(O)**

dataid:growth: indication of what size the approximated end volume of the Dataset is. **(O)**

dataid:usefulness: is used to state to whom the Dataset could be useful, and whether it underpins a scientific publication. **(O)**

[75] `https://ec.europa.eu/programmes/horizon2020/`

[76] `https://ec.europa.eu/research/participants/data/ref/h2020/grants_manual/hi/oa_pilot/h2020-hi-oa-data-mgt_en.pdf`

`dataid:similarData`: a statement on the existence (or absence) of similar data (see also `dataid:relatedDataset`). **(O)**

`dataid:reuseAndIntegration`: information on the possibilities for integration and reuse of the Dataset. **(O)**

`dataid:openness`: General description of how data will be shared. For example embargo periods (if any), outlines of technical mechanisms for dissemination or a definition of whether access will be widely open or restricted to specific groups. In case the Dataset cannot be shared, the reasons for this should be mentioned (e.g. ethical, rules of personal data, intellectual property, commercial, privacy-related, security-related). **(O)**

The running example is structured in a shallow hierarchy with an instance of `dataid:Superset` representing all dataset of a Arabic DBpedia language edition:

```
<dataid_ar.ttl?set=maindataset>
        a                           dataid:Superset ;
        dataid:associatedAgent      <http://wiki.dbpedia.org/dbpedia-association>  ;
        dataid:growth               <dataid_ar.ttl?stmt=growth> ;
        dataid:openness             <dataid_ar.ttl?stmt=openness> ;
        dataid:reuseAndIntegration  <dataid_ar.ttl?stmt=reuseAndIntegration> ;
        dataid:similarData          <dataid_ar.ttl?stmt=similarData> ;
        dataid:usefulness           <dataid_ar.ttl?stmt=usefulness> ;
        dct:hasVersion               <dataid_ar.ttl?version=1.0.0> ;
        dct:language                 <http://lexvo.org/id/iso639-3/ara> ;
        dct:license                  <http://purl.oclc.org/NET/rdflicense/cc-by-sa3.0> ;
        dct:publisher                <http://wiki.dbpedia.org/dbpedia-association> ;
        dct:rights                   <dataid_ar.ttl?rights=dbpedia-rights> ;
        void:subset                 <dataid_ar.ttl?set=long_abstracts_en_uris>,
        <dataid_ar.ttl?set=interlanguage_links> ;
        void:vocabulary             <http://downloads.dbpedia.org/2015-04/dbpedia_2015-10.owl> ;
        dcat:keyword                "maindataset"@en , "DBpedia"@en ;
        dcat:landingPage            <http://dbpedia.org/> ;
        foaf:isPrimaryTopicOf       <dataid_ar.ttl> ;
        foaf:page                   <http://wiki.dbpedia.org/Downloads2015-10> .
```

Listing 5.2: Instance of a Superset

This dataset has no distributions, the data it represents is referred to via its sub datasets. As an example for one of its sub datasets, the following listing exemplifies this difference:

```
<dataid_ar.ttl?set=long_abstracts_en_uris>
        a                       dataid:Dataset, dataid-ld:LinkedDataDataset ;
        dataid:associatedAgent  <http://wiki.dbpedia.org/dbpedia-association> ;
        dataid:qualifiedDatasetRelation   <dataid_ar.ttl?relation=source&target=pages_articles> ;
        dataid:relatedDataset   <dataid_ar.ttl?set=pages_articles> ;
        dct:hasVersion           <dataid_ar.ttl?version=1.0.0> ;
        dct:isPartOf             <dataid_ar.ttl?set=maindataset> ;
        dct:language             <http://lexvo.org/id/iso639-3/ara> ;
        dct:license              <http://purl.oclc.org/NET/rdflicense/cc-by-sa3.0> ;
```

```
dct:title                "long abstracts en uris"@en ;
void:rootResource        <dataid_ar.ttl?set=maindataset> ;
void:triples             232801 ;
void:sparqlEndpoint      <http://dbpedia.org/sparql> ;
dcat:distribution        <dataid_ar.ttl?sparql=DBpediaSparqlEndpoint> ,
    <dataid_ar.ttl?file=long_abstracts_en_uris_ar.ttl.bz2> ,
    <dataid_ar.ttl?file=long_abstracts_en_uris_ar.tql.bz2> ;
dcat:keyword             "long_abstracts_en_uris"@en , "DBpedia"@en ;
dcat:landingPage         <http://dbpedia.org/> ;
sd:defaultGraph          <http://ar.dbpedia.org> ;
foaf:page                <http://wiki.dbpedia.org/Downloads2015-10> .
```

Listing 5.3: A Dataset

An instance of `dataid-ld:LinkedDataDataset` with three distributions, representing two different RDF serialisations (Turtle - .ttl and Quad-Turtle - .tql) as well as an SPARQL endpoint. Furthermore, a source dataset (out of which this dataset was created) is referenced directly with `dataid:relatedDataset`. The actual type of this relation (it being a `dataid:SourceRelation`) will be evident by the qualified version described with the object of `dataid:qualifiedDatasetRelation`. As a Linked Data dataset, properties such as `void:triples` or `sd:defaultGraph` can be applied.

5.2.3 Distribution

The class `dataid:Distribution` provides the technical description of the data, the 'manifestation' of a dataset. In addition, it serves as documentation of how to access the data described (e.g. `dcat:accessURL`), and which conditions apply (e.g. `dataid:accessProcedure`). Every Distribution of a Dataset must contain all data of the Dataset in the format and location described. It may contain additional data exceeding the defined Dataset, for example when describing a service endpoint. Two Distributions of the same Dataset, therefore, must either contain the exact same data (for example in two different serialisations), or one Distribution must completely subsume the other. The Distribution concept, introduced by the DCAT vocabulary, is crucial to be able to automatically retrieve and use the data described in a **DataID** document, simplifying, for example, data analysis. Additional sub classes, to further distinguish how the data is available on the Web, were introduced:

`dataid:SingleFile`: all data of a Dataset is available in a single file.

`dataid:Directory`: the data of a Dataset is represented by the files in a single Directory on a file system.

`dataid:FileCollection`: an arbitrary collection of (different files, not restricted to one file system), which in their accumulation represent the data of a Dataset.

`dataid:ServiceEndpoint`: is a superclass for all service/api endpoints which could provide datasets. For example, REST APIs for databases[77] or SPARQL endpoints for Triplestores.

what is this?

Except for `dataid:SingleFile`, all of these subclasses may need additional semantics to describe them in a useful manner. This is not an objective of **`DataID core`**, further extensions of this ontology will address these issues.

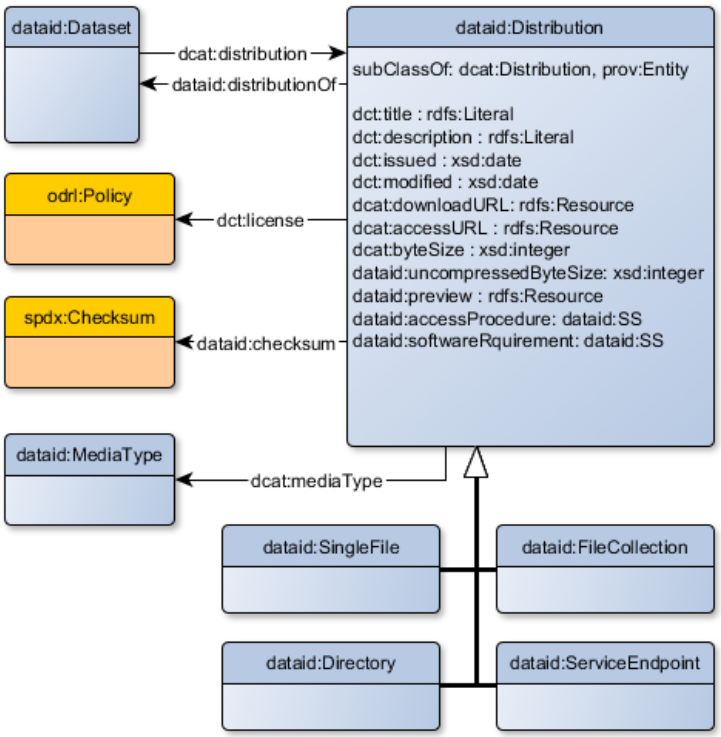


Figure 5.7: Class Distribution

Properties used with `dataid:Distribution`:

`dcat:mediaType`: is restricted to a range of `dataid:MediaType` (see next section 5.2.4). This property is absolutely necessary to automatically digest the content of a Distribution. **(M)**

[77] `https://docs.oracle.com/cloud/latest/mysql-cloud/CSMCS/index.html`

dcat:downloadURL: this property supplies a URL under which the described Distribution can be downloaded directly (without any access procedure or intermediate steps) and completely. **(M if no dcat:accessURL)**

dcat:accessURL: when additional steps are necessary to achieve access to the data of this Distribution (such as authorization, querying or selecting data from a repository), dcat:accessURL is used to provide the initial URL. If the process necessary to retrieve the data is not evident by the content of the web page referred, publishers should make use of property. dataid:accessProcedure. **(M if no dcat:downloadURL)**

dataid:accessProcedure: is used to convey a description of necessary steps, needed to retrieve the data of this Distribution from the dcat:accessURL. This property should not be used in connection with dcat:downloadURL. **(R if no dcat:downloadURL)**

dataid:checksum: the range of this property is restricted to spdx:Checksum [48], providing spdx:checksumValue and spdx:algorithm properties for an exact definition of a checksum. Checksums can be used to validate the correctness of downloaded files, directories and API endpoint responses. **(R)**

dcat:byteSize: The exact size of a distribution in bytes. **(R)**

dataid:softwareRequirement: Some data formats/serialisations are only useful with a particular software product. This statement offers the possibility to name such circumstance. **(O)**

dataid:uncompressedByteSize: Often, files and media streams are compressed to reduce the amount of bytes to be transferred. This optional property provides the means to specify the size of the Distribution in its uncompressed extension. **(O)**

dataid:preview: While the exact format and serialisation of the Distribution is defined by dataid:MediaType, it is often beneficial for publishers and consumers to have a look at the actual format of the (uncompressed) data. This property points out a web resource providing such a preview. **(O)**

dataid:distributionOf: the inverse property of dcat:distribution, pointing back to the Dataset instance this Distribution belongs to. **(O)**

dct:license: see section 5.2.2 for a detailed description. Often, the license used for the Distribution is the same as for the pertaining Dataset. For those cases where this is not true, the use of this property in the context of a Distribution is advised. **(O)**

The first example is an instance of dataid:SingleFile, describing a single RDF file (which contains the whole Dataset) in Turtle syntax, compressed with the

bzip2 compression. It can be downloaded directly (without any intermediate steps), hence the property `dcat:downloadURL` is used to point out the resource on the Web. Since it is a compressed file, the byte size in its compressed and uncompressed state is provided. An instance of `spdx:Checksum` was included, providing the checksum value for this Distribution.

```
<dataid_ar.ttl?file=long_abstracts_en_uris_ar.ttl.bz2>
        a                       dataid:SingleFile ;
        dct:license             <http://purl.oclc.org/NET/rdflicense/cc-by-sa3.0> ;
        dct:publisher           <http://wiki.dbpedia.org/dbpedia-association> ;
        dataid:associatedAgent  <http://wiki.dbpedia.org/dbpedia-association> ;
        dataid:checksum         <dataid_ar.ttl?file=long_abstracts_en_uris_ar.ttl.bz2&checksum=md5>
             ;
        dataid:isDistributionOf <dataid_ar.ttl?set=long_abstracts_en_uris> ;
        dataid:preview          <http://downloads.dbpedia.org/preview.php?file=2015-10_sl_core-
             i18n_sl_ar_sl_long_abstracts_en_uris_ar.ttl.bz2> ;.
        dataid:uncompressedByteSize      186573907 ;
        dcat:byteSize                    33428372 ;
        dcat:downloadURL        <long_abstracts_en_uris_ar.ttl.bz2> ;
        dcat:mediaType          dataid:MediaType_turtle_x-bzip2

<dataid_ar.ttl?file=long_abstracts_en_uris_ar.ttl.bz2&checksum=md5>
        a                   spdx:Checksum ;
        spdx:algorithm      spdx:checksumAlgorithm_md5 ;
        spdx:checksumValue  "2503179cd96452d33becd1e974d6a163"^^xsd:hexBinary .
```

Listing 5.4: A single file Distribution

The second example is an instance of `dataid-ld:SparqlEndpoint`, a sub class of `dataid:ServiceEndpoint` and `sd:Service` which was introduced with the DataID extension for Linked Data (dataid-ld). Additional properties from the SPARQL 1.1 Service Description language are used to further describe the endpoint. As opposed to the previous example, this SPARQL endpoint provides multiple Datasets at once in the context of the original **`DataID`** from DBpedia.

```
<dataid_ar.ttl?sparql=DBpediaSparqlEndpoint>
        a                       dataid-ld:SparqlEndpoint ;
        dataid:associatedAgent  <http://support.openlinksw.com/> ;
        dataid:accessProcedure  <dataid_ar.ttl?stmt=sparqlaccproc> ;
        dct:hasVersion          <dataid_ar.ttl?version=1.0> ;
        dct:license             <http://purl.oclc.org/NET/rdflicense/cc-by-sa3.0> ;
        dcat:accessURL          <http://dbpedia.org/sparql> ;
        dcat:mediaType          <http://dataid.dbpedia.org/ns/mt
        sd:endpoint             <http://dbpedia.org/sparql> ;
        sd:supportedLanguage    sd:SPARQL11Query ;
        sd:resultFormat         <http://www.w3.org/ns/formats/RDF_XML>,
            <http://www.w3.org/ns/formats/Turtle> .
```

Listing 5.5: Distribution of a SPARQL endpoint

5.2.4 MediaType

DCAT does not offer an intrinsic way of specifying the exact format of the content described by a Distribution. While the property `dcat:mediaType` does exist, its expected range `dct:MediaTypeOrExtend` is an empty concept, not extended by DCAT. Therefore, the `dataid:MediaType` was introduced to better qualify this crucial piece of information. The following properties are of interest:

`dataid:typeTemplate`: the IANA media type[78] - also named mime type [49]. This property is of utmost importance for the automatic processing of content. Through its registration with the IANA[79], each type is unambiguously defined and the format of data content can be interpreted in automated processes. **(M)**

`dataid:typeName`: name of the described format or serialisation. **(R)**

`dataid:typeExtension`: a common file extension for the type described (e.g. '.ttl'). **(O)**

`dataid:typeReference`: in some instances, a reference to a useful resource about a type is advisable, to further aid the apprehension of consumer agents (person or software). **(O)**

`dataid:innerMediaType`: with this property, descriptions of nested formats become possible (such as a compressed XML file - '.xml.bz2'), useful in pipeline processing. **(O)**

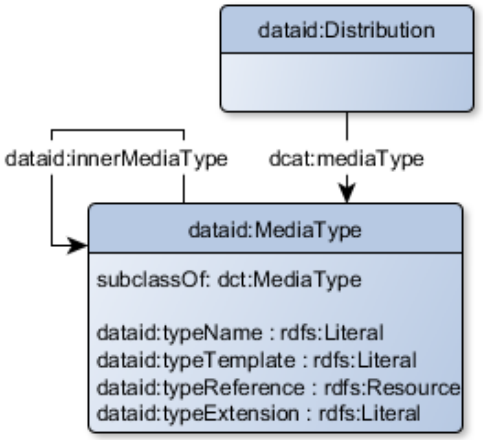


Figure 5.8: Schematic view: A shallow hierarchy of Datasets.

[78] http://www.iana.org/assignments/media-types/media-types.xhtml
[79] http://www.iana.org

The following extract exemplifies the use of these properties: [80]

```
<http://dataid.dbpedia.org/ns/mt#MediaType_turtle_x-bzip2>
        a                       dataid:MediaType ;
        dataid:innerMediaType   <http://dataid.dbpedia.org/ns/mt#MediaType_turtle> ;
        dataid:typeName         "Turtle Bzip2" ;
        dataid:typeExtension    ".bz2" ;
        dataid:typeTemplate     "application/x-bzip2"

<http://dataid.dbpedia.org/ns/mt#MediaType_turtle>
        a                      dataid:MediaType ;
        dataid:typeName        "Turtle" ;
        dataid:typeExtension   ".ttl" ;
        dataid:typeTemplate    "application/x-turtle" ;
        dataid:typeTemplate    "text/turtle" .
```

Listing 5.6: Example of a complex media type

5.2.5 Agent

An Agent is something or someone that bears some form of responsibility for an Entity or activities which create, transform or manage Entities in some way. Agents are real or legal persons, groups of persons, programs, organisations etc. The class `dataid:Agent` subsumes both agent concepts of PROV-O and FOAF ontologies, to further incorporate PROV-O into the context of DCAT (which uses `foaf:Agent` to portray this concept).

The attributes of the FOAF vocabulary[81] are used to describe aspects such as name and e-mail address of a person. In addition, the following properties were introduced in **DataID core**:

`dataid:hasAuthorization`: the inverse property of `dataid:authorizedAgent`, pointing out an Authorization which grants an Agent some kind some authority over Entities. This is explained in detail in the example on Authorizations in the next section 5.2.6. **(R)**

`dataid:identifier`: often, Agents have already unique identifiers somewhere on the Web (in addition to the URI used in the context of a **DataID** document, such as URI, ORCID[82] or researcherId[83]). This property points out additional identifiers (see section 5.2.9). **(O)**

[80] note: the namespace `http://dataid.dbpedia.org/ns/mt#` for common MediaTypes is used on a preliminary basis)
[81] `http://xmlns.com/foaf/spec/`
[82] `http://orcid.org`
[83] `http://www.researcherid.com`

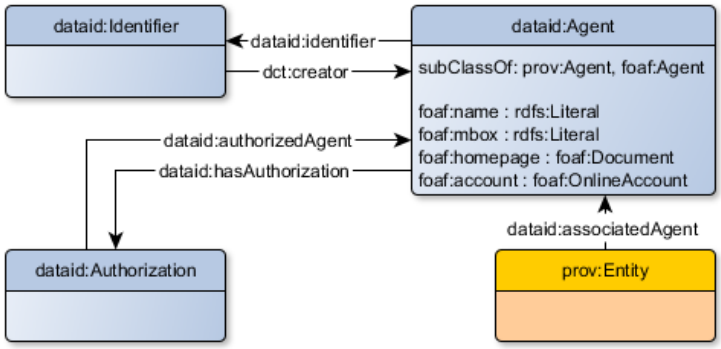


Figure 5.9: Schematic view: A shallow hierarchy of Datasets.

This example of an Agent portrays the DBpedia association[84]:

```
<http://wiki.dbpedia.org/dbpedia-association>
        a                       dataid:Agent ;
        dataid:hasAuthorization  <dataid_ar.ttl?auth=creatorAuthorization> ;
        foaf:homepage            <http://dbpedia.org> ;
        foaf:mbox                "dbpedia@infai.org" ;
        foaf:name                "DBpedia Association" .
```

Listing 5.7: Example of an organisation

5.2.6 Authorization

One objective of **`DataID core`** is the detailed expression of the relations between Agents and Entities. To qualify these relations (summarised under the property `dataid:associatedAgent`) AgentRoles have to be assigned to the involved Agents (such as Maintainer, Publisher etc.). This is achieved by the class `dataid:Authorization`, which is a sub class of `prov:Attribution`, a qualification of the property `prov:wasAttributedTo`. It basically states, which AgentRole(s) (pointed out with `dataid:authorityAgentRole`) an Agent (via `dataid:authorizedAgent`) has, regarding a certain collection of Entities (`dataid:authorizedFor`). This mediator is further qualified by an optional period of time for which it is valid and access restrictions by the Entities themselves, allowing only specific Authorizations to exert influence over them (see `dataid:needsSpecialAuthorization`).

Chapter 6 contains a detailed example on Authorizations, to deepen the understanding of this concept as well as to provide suitable use cases.

[84] `http://wiki.dbpedia.org/dbpedia-association`

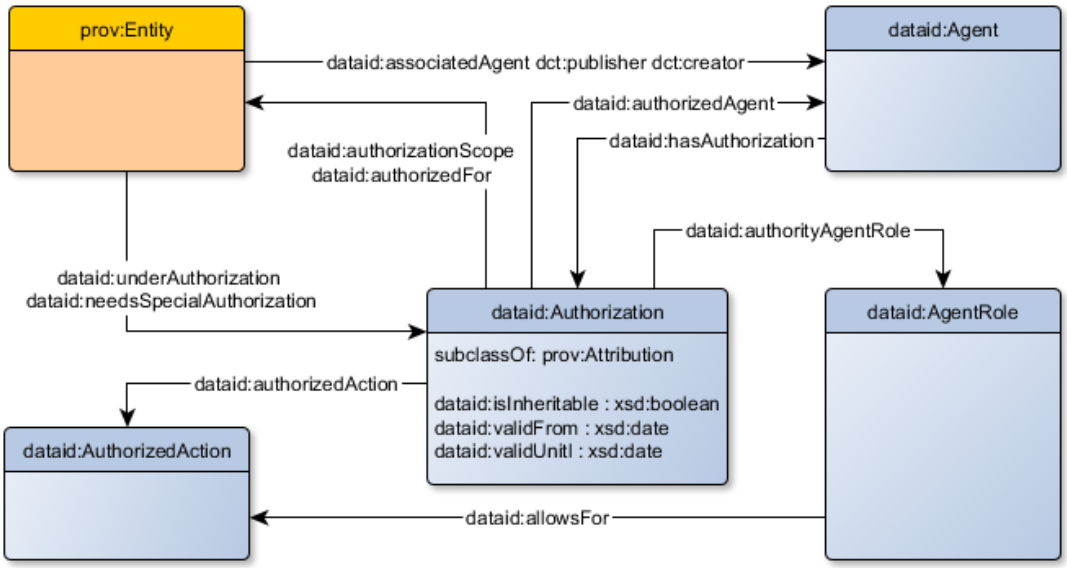


Figure 5.10: Schematic view: A shallow hierarchy of Datasets.

`dataid:authorizedAgent`: each Authorization shall have at least on associated Agent, which are pointed out via this property (sub property of `prov:agent`). **(M)**

`dataid:authorityAgentRole`: provides the AgentRole the Agent(s) of this Authorization are assigned with (sub property of `prov:hadRole`). **(M)**

`dataid:authorizedFor`: this property points out those Entities for which this Authorization is valid (sub property of `dataid:authorizationScope`). **(M)**

`dataid:authorizedAction`: an AgentRole entails the right (or responsibility) for Agents to execute a collection of defined AuthorizedAction(s). This property can be inferred by a chain of properties (OWL property chain axiom): `dataid:authorityAgentRole` / `dataid:allowsFor`. **(O)**

`dataid:isInheritable`: indicates whether an Authorization is transferable when changing versions of a **`DataID`**. Thus, keeping Agent, AgentRole and Actions in place for the updated versions of all involved Entities. **(O)**

`dataid:validFrom`: defining the temporal beginning (inclusive) of an Authorization (the time from which on out the axioms of an Authorization are valid). **(O)**

`dataid:validUntil`: defining the temporal ending (exclusive) of an Authorization (the time from which on out the axioms of an Authorization are no longer valid). **(O)**

The property `dataid:authorizationScope` is an abstract super property of `dataid:authorizedFor`, pointing to all referred and inferred Entities which are under a certain Authorization. Triples with this predicate are inferred by its sub properties `dataid:authorizedFor` and the virtual properties `dataid:authorizationChain1` to `dataid:authorizationChain9`. An exact explanation of Authorizations and all involved properties is accompanying the extended example on Authorizations at the end of this chapter (see section 5.3).

Furthermore, property `dataid:underAuthorization` is the inverse of `dataid:authorizationScope` and a sub property of `prov:wasAttributedTo`, pointing out the qualification instance which qualifies the relation `dataid:associatedAgent`. Its sub property `dataid:needsSpecialAuthorization` was introduced to restrict the reach of Authorizations, to the exclusion of those Authorizations, not referenced via this property (or simply: if an Entity has a `dataid:needsSpecialAuthorization` instance, all Authorizations without this referral are impotent).

The following snippet provides a simple example of two AgentRoles being assigned to two Agents for a DataId instance (and thereby for every Entity involved in this **`DataID`**- see chapter 6 for more).

```
<http://wiki.dbpedia.org/dbpedia-association/persons/Freudenberg>
        a                       dataid:Agent ;
        dataid:hasAuthorization  <dataid_ar.ttl?auth=maintainerAuthorization> ;
        foaf:mbox               "freudenberg@informatik.uni-leipzig.de" ;
        foaf:name               "Markus Freudenberg"                    ;
        dataid:identifier       <http://www.researcherid.com/rid/L-2180-2016> .

<dataid_ar.ttl?auth=maintainerAuthorization>
        a                         dataid:Authorization ;
        dataid:authorityAgentRole  dataid:Maintainer ;
        dataid:authorizedAgent     <http://wiki.dbpedia.org/dbpedia-association/persons/Freudenberg>
             ;
        dataid:authorizedFor       <dataid_ar.ttl> .

<http://wiki.dbpedia.org/dbpedia-association>
        a                       dataid:Agent ;
        dataid:hasAuthorization  <dataid_ar.ttl?auth=creatorAuthorization> ;
        foaf:homepage           <http://dbpedia.org> ;
        foaf:mbox               "dbpedia@infai.org" ;
        foaf:name               "DBpedia Association" .

<dataid_ar.ttl?auth=creatorAuthorization>
        a                         dataid:Authorization ;
        dataid:authorityAgentRole  dataid:Creator ;
        dataid:authorizedAgent     <http://wiki.dbpedia.org/dbpedia-association> ;
        dataid:authorizedFor       <dataid_ar.ttl> .
```

Listing 5.8: Example of an organisation

5.2.7 AuthorizedAction & AgentRole

The AgentRole assigned to an Agent in the context of an `dataid:Authorization` is defined only by the property `dataid:allowsFor`, pointing out AuthorizedActions it entails. A `dataid:AuthorizedAction` shall either be a dataid:EntitledAction, representing all AuthorizedActions an Agent could take, or the AuthorizedActions an Agent has to take (`dataid:ResponsibleAction`).

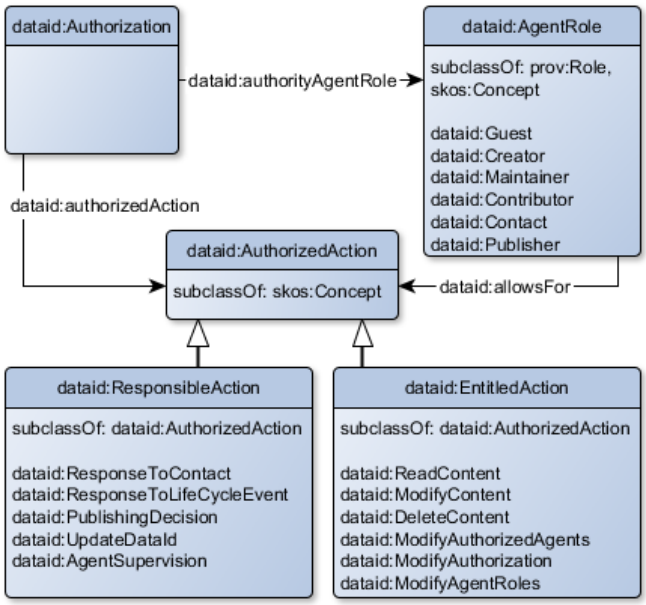


Figure 5.11: Schematic view: A shallow hierarchy of Datasets.

AuthorizedActions and AgentRoles defined in this ontology are only examples of possible implementations, reflecting a common environment of a File or Document Management System. They can be replaced to fit the use case at hand. Implementing them as a `skos:ConceptScheme`[85] offers additional semantics, for example in determining which AgentRole can override AuthorizedActions initiated by Agents with other AgentRoles for the same Entity. The following descriptions will shortly introduce each individual, predefined in the **`DataID core`** ontology.

Agent Roles:

Creator: Creator of the resource. An AgentRole that is credited with the main part in the initial creation of the resource.

[85] `https://www.w3.org/TR/skos-reference/#ConceptScheme`

Contact: An Agent that can be contacted for general requests about the resource.

Contributor: Contributor to the resource. An Agent that was involved in creating or maintaining the resource but does not have the main part in this activity.

Guest: A visitor or anonymous Agent has only read rights on public documents.

Maintainer: Maintainer of the Dataset. An Agent that ensures the technical correctness, accessibility and up-to-dateness of a Dataset.

Publisher: Publisher of the Dataset. An Agent that makes the Dataset accessible online on a server or repository without necessarily being involved in its creation.

Responsible Actions:

ResponseToContact: The responsibility to respond to contact attempts by external Agents. A contact point for the Entity.

ResponseToLifeCycleEvent: The responsibility to manage changes and react to bugs and issues that are reported concerning a Dataset.

PublishingDecision: The responsibility to decide if an Entity should be published.

UpdateDataId: The responsibility to update dataset metadata.

AgentSupervision: The responsibility to supervise other Agents.

Entitled Actions:

ReadDataId: read the DataID dataset metadata.

ReadContent: read the content of an Entity.

ModifyContent: modify the content of an Entity.

DeleteContent: delete some content of an entity.

ModifyAuthorizedAgents: modify which Agents are authorized on certain Entities.

ModifyAuthorization: modify an Authorization.

ModifyAgentRoles: modify the role of Agents on certain Entities.

This example is not part of the running example, but has been lent from the **DataID core** ontology document itself. Here the AgentRole 'Contact' is defined in the context of a `skos:ConceptScheme`.

```
dataid:Contact
        a owl:NamedIndividual, dataid:AgentRole;
        rdfs:isDefinedBy <http://dataid.dbpedia.org/ns/core#> ;
        dataid:allowsFor dataid:ReadDataId;
        dataid:allowsFor dataid:ModifyContent;
        dataid:allowsFor dataid:ReadContent;
        dataid:allowsFor dataid:ResponseToContact;
        rdfs:comment "Contact agent. An agent that can be contacted for general requests about the
             resource."@en ;
        skos:prefLabel "contact"@en ;
        skos:inScheme dataid:AgentRoleScheme ;
        skos:broader dataid:Publisher, dataid:Maintainer .
```

Listing 5.9: Example of an organisation

5.2.8 DatasetRelationship

A DatasetRelationship is a qualification of the generic property `dataid:relatedDataset` (which is a sub-property of `dct:relation`). The `dataid:DatasetRelationship` is a subclass of `prov:EntityInfluence` and is defined by three properties:

`dataid:datasetRelationRole`: specifying the role (or type) of this relationship, defining the exact role the 'target' Dataset takes in regard to the 'origin' dataset of this relationship.

`dataid:qualifiedRelationOf`: the inverse property of `qualifiedDatasetRelation` is pointing out the origin Dataset of this qualification.

`dataid:qualifiedRelationTo`: the target Dataset of this qualification.

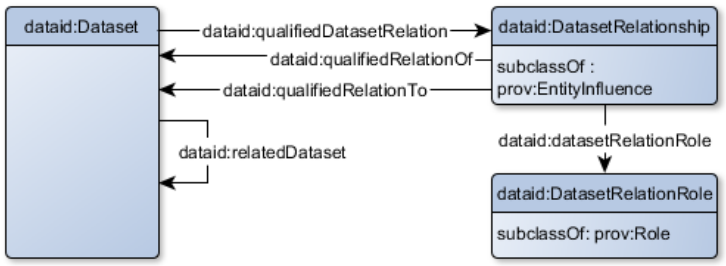


Figure 5.12: Schematic view: A shallow hierarchy of Datasets.

The class `dataid:DatasetRelationRole` is not further qualified in the context of **`DataID core`**, which could be done in an extension to **`DataID core`**, similar to `dataid:AgentRole`. Some instances of this class are already provided:

GenericRelation: specifies a `dataid:DatasetRelationship` between two datasets which have a relation of a unknown quality (such a relation is equivalent to direct `dct:relation` property).

DerivatRole: specifies a `dataid:DatasetRelationship` where one dataset points out a second dataset, which is a derivat of the first.

SourceRole: specifies a `dataid:DatasetRelationship` where the origin dataset is created by transforming/collecting data from the target dataset.

CopyRole: specifies a `dataid:DatasetRelationship` where the origin dataset is an exact copy of the target dataset (e.g. when republished under a different domain).

SimilarityRole: specifies a `dataid:DatasetRelationship` where the origin dataset has a significant similarity to the target dataset (without any assertion as to dimension of similarity).

In the example, the Wikipedia source dataset (named 'pages_articles') of a given DBpedia dataset is referred to with the help of this concept:

```
<dataid_ar.ttl?relation=source&target=pages_articles>
        a                           dataid:DatasetRelationship ;
        dataid:datasetRelationRole  dataid:SourceRole ;
        dataid:qualifiedRelationOf  <dataid_ar.ttl?set=long_abstracts_en_uris> ;
        dataid:qualifiedRelationTo  <dataid_ar.ttl?set=pages_articles> .
```

Listing 5.10: Example of an organisation

5.2.9 Identifier

The class `dataid:Identifier` uniquely identifies any resource (incl. Entities and Agents), given an identifier as a literal and a corresponding `datacite:IdentifierScheme` (e.g. ORCID, ResearcherID etc.). Typically an organisation is responsible for issuing and managing Identifiers described with this concept, which can be referred to with `dct:creator`. **`DataID core`** adopted this approach from Datacite ontology (see **??**) to provide a schematic way of adding additional, existing identifiers to Entities and Agents referenced in a **`DataID`** document.

add something about refining the datacite ontology

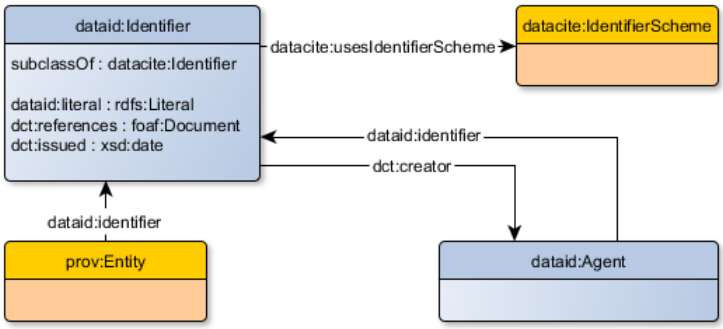


Figure 5.13: Schematic view: A shallow hierarchy of Datasets.

`dataid:literal`: the identifier as literal (e.g. a URI as literal). **(M)**

`datacite:usesIdentifierScheme`: an IdentifierScheme defines (among other attributes) a pattern against which the literal of an identifier is validated. Thereby the validity of an identifier is tested. **(M)**

`dct:references`: often identifier agencies have web presentations for their identifiers (e.g. ORCID[86]). Such a website can be referenced with this property. **(O)**

`dct:creator`: can be used to identify the identifier agency, responsible for an identifier and pertaining scheme. **(O)**

```
<http://www.researcherid.com/rid/L-2180-2016>
        a                              dataid:Identifier ;
        dataid:literal                 "L-2180-2016" ;
        dct:issued                     "2016-08-01"^^xsd:date ;
        dct:references                 <http://www.researcherid.com/rid/L-2180-2016> ;
        datacite:usesIdentifierScheme  datacite:researcherid .
```

Listing 5.11: Example of an organisation

5.2.10 SimpleStatement

The concept dataid:SimpleStatement is intended as a tool for conveying a statement, definition or point of view about a certain topic. Using either a simple literal (using `dataid:literal`) to provide a quotation or by a referencing a web resource providing or representing the statement in any medium (picture, text, video etc.). This class is a sub class of `prov:Entity` and implements in addition the following Dublin Core classes: `dct:ProvenanceStatement`,

[86] `http://orcid.org/0000-0002-1825-0097`

`dct:RightsStatement, dct:Standard .` With this measure, it is possible to attach provenance information onto instances of this concept and to use `dataid:SimpleStatement` as range of `dct:rights` and its sub-properties, `dct:provenance`, `dct:conformsTo` and others.

This reification approach with an intermediate resource was chosen to cover as many scenarios as possible including many edge cases which do not have to be modelled explicitly. To provide a minimum of structure for textual statements, as well as providing an resource onto which provenance information could be it easy to attach provenance information to a statement.

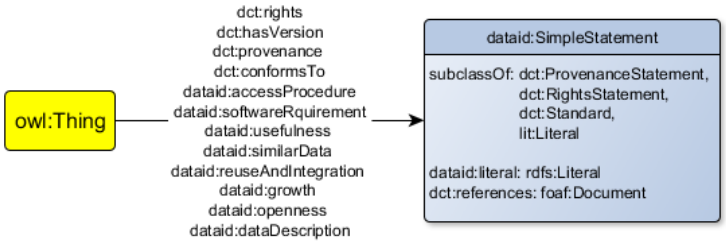


Figure 5.14: Schematic view: A shallow hierarchy of Datasets.

`dataid:literal`: a textual statement 'from humans for humans'. **(R)**

`dct:references`: the alternative reference to a web resource containing the human readable statement. **(O)**

Two instances from the running example demonstrating the different usage scenarios of this concept. The first is the official rights statement of the DBpedia association, while the second is the access procedure for the DBpedia SPARQL endpoint, where `dct:reference` is pointing out the SPARQL 1.1 specification (as a web page specifying the means of querying the endpoint).

define SPARQL somewhere

```
<dataid_ar.ttl?rights=dbpedia-rights>
        a                dataid:SimpleStatement ;
        dataid:literal  """DBpedia is derived from Wikipedia and is distributed under the same
             licensing terms as Wikipedia itself. As Wikipedia has moved to dual-licensing, we also
             dual-license DBpedia starting with release 3.4. Data comprising DBpedia release 3.4 and
             subsequent releases is licensed under the terms of the Creative Commons Attribution-
             ShareAlike 3.0 license and the GNU Free Documentation License. Data comprising DBpedia
             releases up to and including release 3.3 is licensed only under the terms of the GNU Free
              Documentation License."""@en .

<dataid_ar.ttl?stmt=sparqlaccproc>
        a                dataid:SimpleStatement ;
        dct:references    <https://www.w3.org/TR/sparql11-overview/> ;
        dataid:literal  "An endpoint for sparql queries: provide valid queries."@en .
```

Listing 5.12: Example of an organisation

5.3 Complex Example on Authorizations

I decided to provide a more prolific example on the subject of Authorizations, since this concept is of more complex nature. In particular, the impact of `dataid:authorizationScope` with its sub-properties is difficult to understand at first sight.

The property `dataid:authorizationScope` has the role of an abstract super property, pointing out all referred and inferred Entities under a given Authorization and is usually not instantiated in a **`DataID`** graph. It can be inferred directly by the existence of `dataid:authorizedFor`, which is used to reference Entities to which an Authorization applies and all its rules and restrictions are tailored for.

The following axioms for the transitive property `dataid:authorizationScope` would be desirable to extend the influence of an Authorization along any property path combined of `foaf:primaryTopic`, `void:subset` and `dcat:distribution`, initiated by an instance of `dataid:authorizedFor`.

$$\begin{aligned} foaf : primaryTopic &\sqsubseteq dataid : authorizationScope \\ void : subset &\sqsubseteq dataid : authorizationScope \\ dcat : distribution &\sqsubseteq dataid : authorizationScope \\ dataid : authorizedFor &\sqsubseteq dataid : authorizationScope \end{aligned}$$

To not hijack foreign ontologies [6] (i.e. DCAT, VOID or FOAF), a series of properties (`dataid:authorisationChain1 - dataid:authorisationChain9`) were introduced to simulate this behaviour with the help of the OWL property chain axiom. needs link

Properties `dataid:authorisationChain1` to `dataid:authorisationChain6`:

$$\begin{aligned} foaf : primaryTopic \circ dcat : distribution &\sqsubseteq dataid : authorizationScope \\ foaf : primaryTopic \circ void : subset &\sqsubseteq dataid : authorizationScope \\ void : subset \circ dcat : distribution &\sqsubseteq dataid : authorizationScope \\ void : subset \circ void : subset &\sqsubseteq dataid : authorizationScope \\ void : subset \circ void : subset \circ dcat : distribution &\sqsubseteq dataid : authorizationScope \\ foaf : primaryTopic \circ void : subset \circ dcat : distribution &\sqsubseteq dataid : authorizationScope \end{aligned}$$

With these properties all subsequent Entities connected to the origin Entity (the Entity referenced from an Authorization with `dataid:authorizedFor`), are under the influence of this Authorization, connected with it via the transitive `dataid:authorizationScope`. One exception remains: those Entities second in line after the origin Entity are skipped in this progression. Properties

`dataid:authorisationChain7` to `dataid:authorisationChain9` are solving this issue:

$$dataid : authorizedFor \circ foaf : primaryTopic \sqsubseteq dataid : authorizationScope$$
$$dataid : authorizedFor \circ void : subset \sqsubseteq dataid : authorizationScope$$
$$dataid : authorizedFor \circ dcat : distribution \sqsubseteq dataid : authorizationScope$$

For example: if a knowledge base (KB) holds some statements, such as:

```
ex:someAuthorization    dataid:authorizedFor          ex:DataId .
ex:DataId               foaf:primaryTopic             ex:RootDataset .
ex:RootDataset          void:subset                   ex:DatasetC .
```

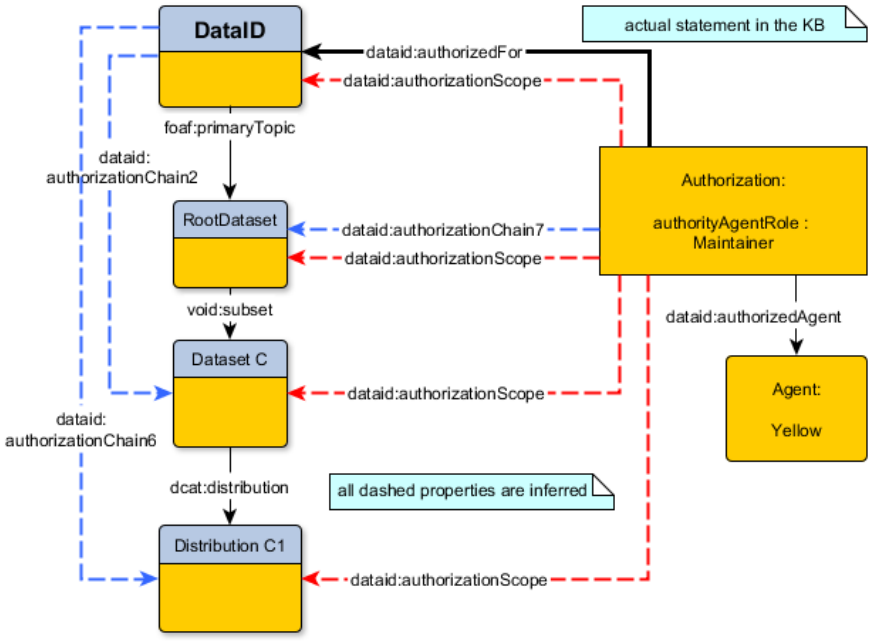
we can infer the following statements:

```
ex:someAuthorization    dataid:authorizationScope     ex:DataId .
ex:someAuthorization    dataid:authorizationScope     ex:RootDataset .
ex:someAuthorization    dataid:authorizationScope     ex:DatasetC .
```

by inferring these statements first:

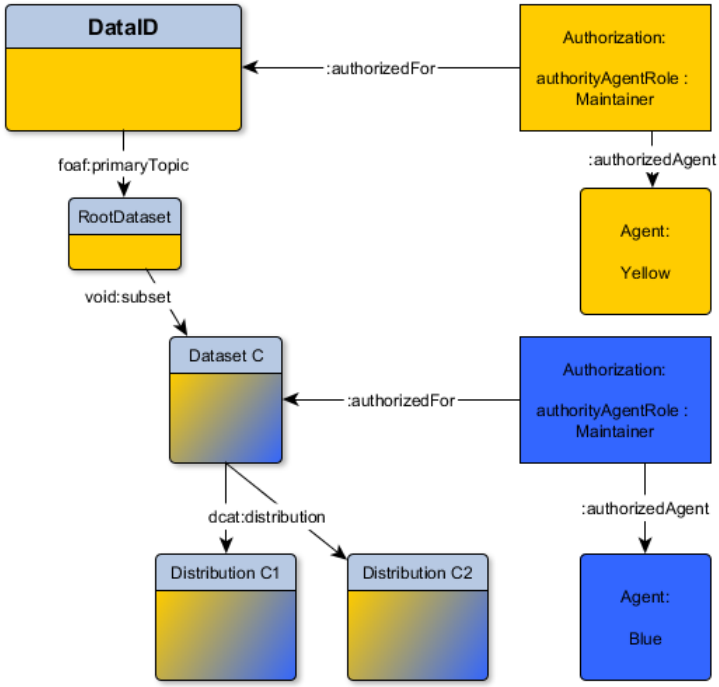
```
ex:DataId               dataid:authorizationChain2    ex:DatasetC .
ex:DataId               dataid:authorizationChain6    ex:DistributionC1 .
ex:someAuthorization    dataid:authorizationChain7    ex:RootDataset .
```

With these auxiliary properties the inference of `dataid:authorizationScope` along the depicted property paths becomes feasible:



Here the Authorization for Agent Yellow is not only valid for the **DataID** entity, referred to via `dataid:authorizedFor`. By inferring additional statements of this kind, the scope of this Authorization is extended to every Dataset and Distribution connected via `foaf:primaryTopic`, `dcat:distribution` and `void:subset`. By this means, extending the influence (or scope) of an Authorization over multiple Entities without having to point out all of them with `dataid:authorizedFor` is realised, without changing the definitions of the external properties involved, or an inclusion of rule-based axioms (such as SWRL[87]).

The automatic extension of an Authorization has also its drawbacks. By introducing multiple Authorizations in the context of a **DataID** document, providing the same AgentRole for an Entity, the author can encounter unintended behaviours. In this example the previous context is enriched by introducing an additional Agent Blue:

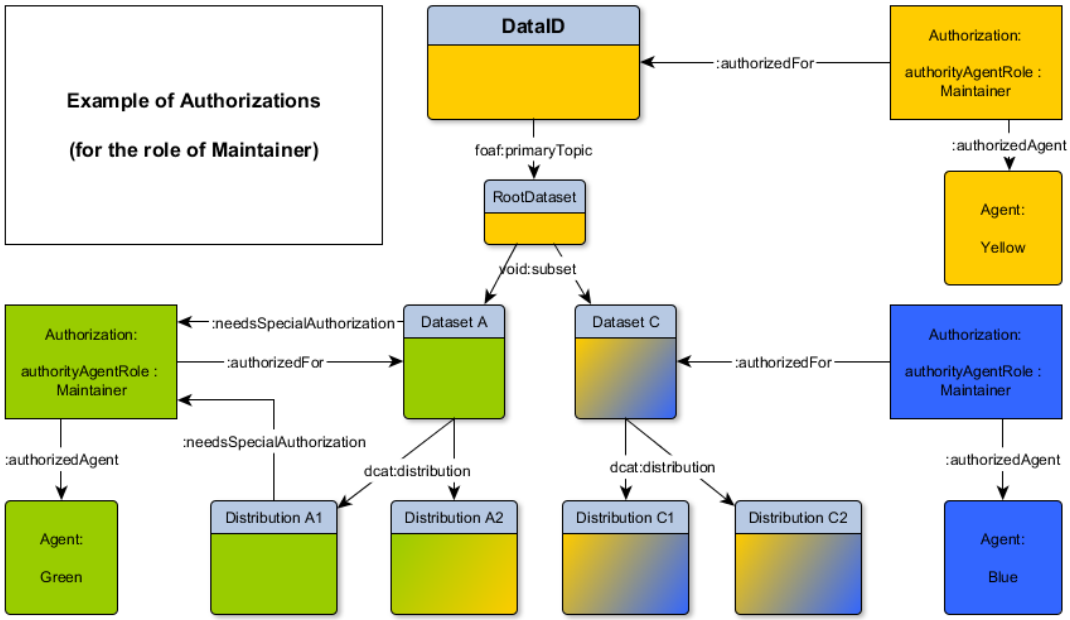


DatasetC (and all its Distributions) has two Maintainers, both equally permitted to wield AuthorizedActions as defined by the definition of `dataid:Maintainer`. This behaviour may or may not be intended by the author. To provide the means for restricting Entities to specific Authorizations, the property `dataid:needsSpecialAuthorization` was introduced. This sub-property of `dataid:underAuthorization` (the inverse of `dataid:authorizationScope`) allows to point out those Authorizations with sufficient importance to exert their

[87] `https://www.w3.org/Submission/SWRL/`

authority over an Entity, to the exclusion of other Authorizations referenced via `dataid:authorizationScope`.

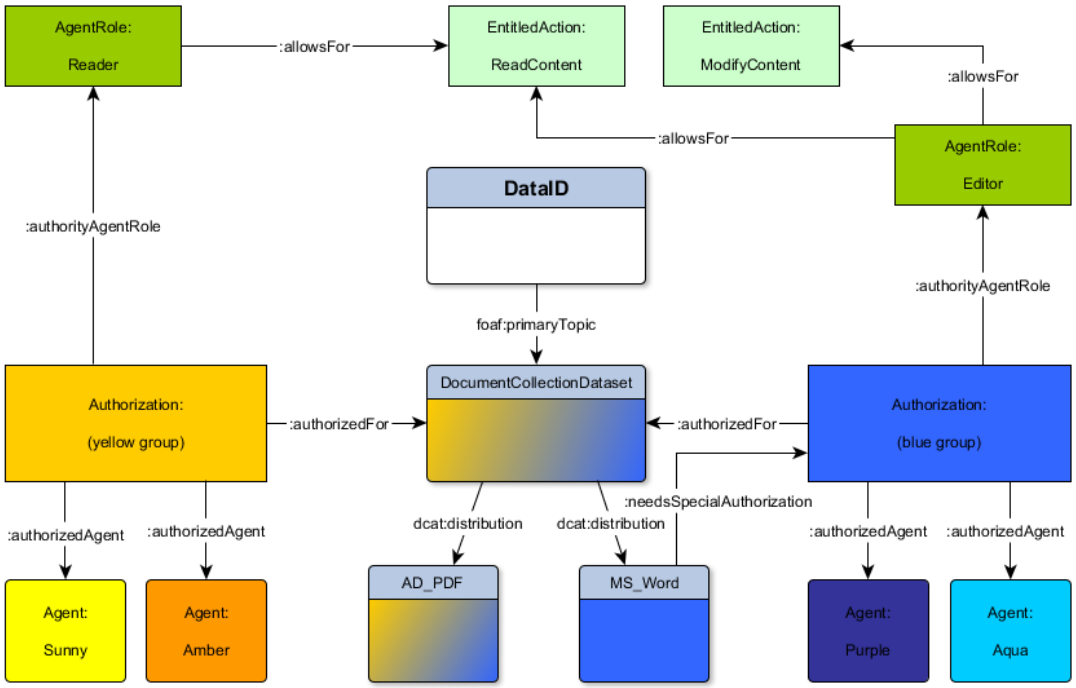
The following example again expands the already known scenario, by introducing a third Authorization for Agent Green:



While DatasetA and DistributionsA1 and A2 are under the Authorizations of Agent Yellow and Agent Green, only DistributionA2 will be maintained by both Agents. DatasetA and DistributionA1 require specifically the Authorization of Agent Green for the purpose of providing the AgentRole of Maintainer.

This mechanism is useful when introducing different levels of privacy into the domain of, for example, a Document Management System (DMS). Two groups of users are specified: The first group (yellow group) should only be able to read the content of a given collection of documents, while the second group (blue group) is also allowed to modify these documents. Therefore, defining two new AgentRoles is advisable. AgentRole 'Reader' can only read the content of Entities available to it, while the 'Editor' allows also for modifying the content. These AgentRoles are linked to via `dataid:authorityAgentRole` from the respective Authorizations of the two groups (`dataid:authorizedAgent` points out the members of a group).

Both Authorizations are authorised for the same document collection (Dataset) and its Distributions as PDF and MS_Word versions of the same content in the DMS. Since the MS_Word version of the documents is used for editing



the content, while its PDF counterpart is the publishing version, it is sensible to allow only the Editors (blue group) access to the MS_Word Distribution by using `dataid:needsSpecialAuthorization`.

6 Publishing Data with DataID

6.1 Best Practices

Best practices on any kind of methodology or problem have become a ubiquitous presence in the current landscape of the World Wide Web. Be it a Ted Talk[88] on how to live to be 100[89] or how to get accurate results when using machine learning techniques[90].

Astonishingly, when it comes to publishing data in a widely accepted manner, only a hand full of comprehensive best practices exist. Most of these best practices, workflows or checklists are further constricted to a certain field of research, methodology or data type:

Methodological Guidelines for Publishing Government Linked Data "[...] a preliminary set of methodological guidelines for generating, publishing and exploiting Linked Government Data" [50]

Best Practices for Publishing Linked Data "[...] a series of best practices designed to facilitate development and delivery of open government data as Linked Open Data." [51]

Key components of data publishing "From an assessment of the current data-publishing landscape, we highlight important gaps and challenges to consider, especially when dealing with more complex workflows and their integration into wider community frameworks." [52]

add one more add names

There are two not quiet distinct categories of best practices on data publishing.

Workflows introduce a specific order of activities, the flow of artefacts between them as well as agents and their roles in these processes. Workflows are often summarised in so called Data Lifecycles (or Data Engineering Lifecycle).

[88] https://www.ted.com

[89] https://www.ted.com/talks/dan_buettner_how_to_live_to_be_100

[90] http://machinelearningmastery.com/machine-learning-checklist/

The LOD2 Lifecycle of Linked Data [53] - used by the ALIGNED project (see fig. 1.1) - is an portrayal of such a workflow in a domain specific environment.

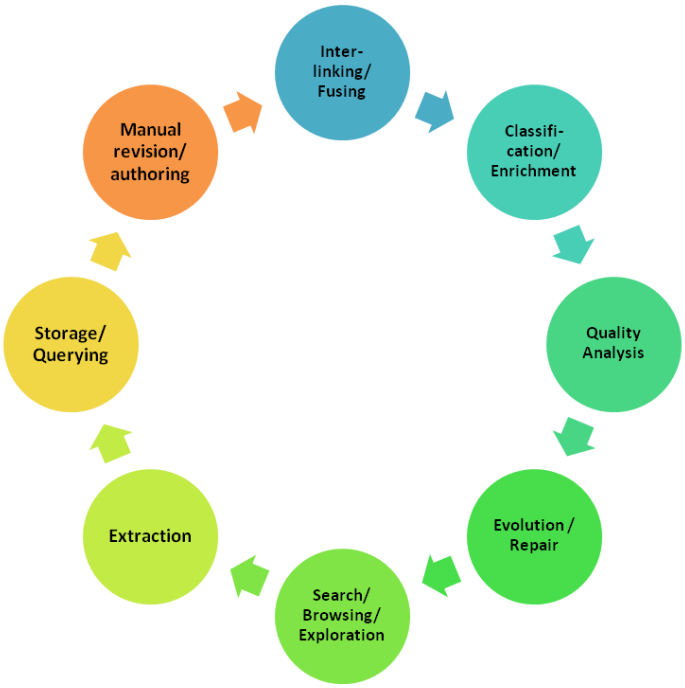


Figure 6.1: The LOD2 Lifecycle of Linked Data [53]

Many depiction of Data Engineering Lifecycles have domain-independent similarities. Villazón-Terrazas et al. presented a more generalized version of such a cycle, which I am going to adopt in the context of this work.

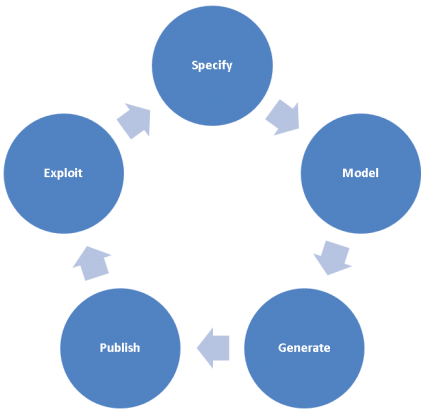


Figure 6.2: The Government Linked Data Lifecycle [50]

A concise summary, tracing the bubble graph above:

- Specify
  - analysis of data sources (which data (slices) are useful) or previous version of the dataset
  - Select/Design Identifier scheme user to identify records
- Model
  - search/create suitable schema/vocabulary on which to base the datasets
- Generate
  - transform the data sources
  - clean (and validate) the result
- Publish
  - improve discoverability (choose suitable data portal, announce release at different venues)
  - publish the created datasets
  - publish the pertaining metadata
- Exploit
  - different consumer side activities (browsing, integrating, analysing etc.)
  - collect feedback from consumers, improve data

Checklists are collections of general advises, how-tos and precautions on publishing data without a particular ordering of items.

The most comprehensive collection of best practices about publishing data has emerged with the establishment of the 'Data on the Web' W3C Working Group[91] and their best practices document [3] (released as recommendation candidate by the time of writing).

This set of recommendations touches upon most of the crucial issues when publishing data (with the noted exception of dissemination tasks). I endorse these 35 best practices to their full extend and recommend following all of the suggestions made. Datasets published with adhering to the suggestions made would reap all benefits towards which these best practices were aimed: *Reuse, Comprehention, Linkability, Discoverabilty, Trust, Access, Interoperability and Processability*.

check phrasing

[91] https://www.w3.org/2013/dwbp/wiki/Main_Page

In this chapter I want to build on this foundation, by expanding on metadata composition and deployment with **DataID** (section 6.2). In addition, I want to contribute to the discussion on publishing data, with a specialised checklist for publishing Linked Data datasets (section 6.3), based on the experiences I have accumulated when publishing official DBpedia releases.

specialised?

6.2 Composing and Publishing DataID based Metadata

The Data Engineering Lifecycle depicted in Figure 6.2 does not only apply to the data being generated and published. It is also a valid depiction of a lifecycle for metadata which is co-evolving parallel to the data it is portraying.

In the case of **DataID** metadata, the tasks necessary to eventually publish a metadata document are outlined in this best practice. I will apply the five stages of the Data Engineering Lifecycle to the process of creating **DataID** documents.

Specify use case requirements for metadata Based on the analysis of data sources, external requirements and advise collected in the run up to the effort of creating a dataset, a set of requirements for dataset metadata has to be outlined as well. This is a highly use case specific process. Some fundamental question which might guide this process are as follows:

- Who is the intended audience of the data (is it a document intended for reading by humans etc.)?
- How will the data usually be consumed?
- How deep is the need for provenance information?
- What are the legal terms under which the data is published?
- What structure will the data have???
- Will there be multiple versions of this dataset
- What type of data am I creating (e.g. Linked Data, table based data etc.)
- How was data published by my organisation/partners until now?
- What would be the best solution for my costumers?
- what standards or vocabularies were used by the source datasets

revise

Model a DataID ontology and application profile A crucial step when implementing a **DataID** based metadata solution is the decision on which **DataID** extension ontologies to use, which is a domain and problem dependent process.

Deciding on which combination of DataID ontologies to use for a Dataset description is It may be necessary to add additional properties on top of the provided metadata properties.

For example, a DataID based ontology for LOD Datasets dealing with multidimensional data, may look schematically like this: (importing DataID core and the extensions for Linked Data and Statistics, as well as some additional properties only used in the use case at hand.)

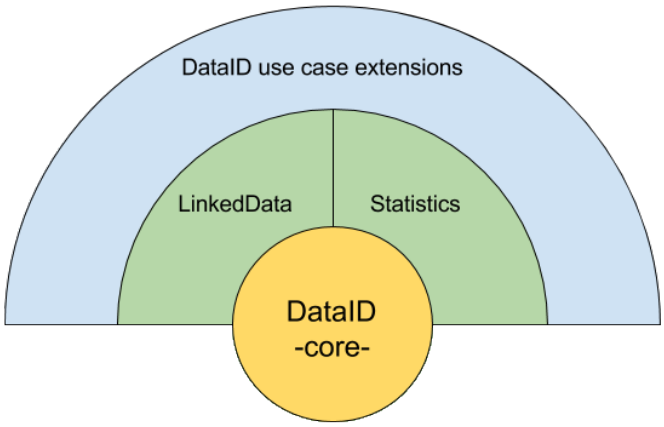


Figure 6.3: Example of combining multiple DataID ontologies

It may be necessary to add additional properties combined in a use case extension to satisfy all requirements. This process will be the focus of the Data Management Plan use case presented in chapter 7. In general, the usual recommendations for ontology engineering apply, when creating a use case extension for **DataID**.

add reference

- follow an established methodology when creating an ontology [**todo**]
- reuse well established ontologies where possible (such as W3C recommended ontologies)
- create new classes based on more general concepts to harness additional functionality and increase interoperability (e.g. `prov:Entity`)
- make use of established tooling chains for modelling ontologies (such as Protege)

character an url

- provide sufficient documentation to ease understanding and reuse

In addition, I recommend the following:

- keep the import portfolio as slim as possible - only import extensions or other ontologies which are really needed for the use case at hand to avoid semantic overlap
- separate ontology from application profile - don't use the extension ontology for specifying use case specific restrictions, enter those into a second document. I recommend using the Shapes Constraint Language (SHACL) [47] from the RDF Data Shapes W3C Working Group[92] which is available as a working draft at the time of writing. This measure will aid the effort for **interoperability**.
- specify mappings to other metadata formats at this stage to detect missing requirements - reasons for this step might be: already existing metadata files or it is needed for specific tasks such as a CKAN profile (section 3.1.3) for publishing datasets on a data portal

shacl

Generate DataID documents Creation of metadata documents is usually a task executed after the creation and before the publication of datasets. Yet, there are multiple reasons to not wait until this stage to gather the necessary data:

- describing source datasets is a task best done at the time before the extraction process
- the dataset creation process might take a long time, depending on the use case. Metadata for temporary result datasets can be useful for further extraction/transformation steps or to keep track of activities and agents involved (which is useful provenance metadata)

When publishing a series of similar or versioned datasets, the automatic creation of **DataID** documents becomes unavoidable. Integrating the creation of metadata as part of an automated workflow environment or ETL Framework (e.g. Unified Views [54]) would be an obvious means of solving this issue.

As for the data generated, **DataID** metadata needs a validation to make sure of its correctness. While a syntactical validation of an RDF document is necessary step it is not sufficient. Tools for a semantic validation of an RDF graphs against the ontology specified in the previous step are available (e.g. RDFUnit [55]).

[92] https://www.w3.org/2014/data-shapes/wiki/Main_Page

Publish DataIDs alongside the datasets Since dataset metadata is so indispensable for most consumers, the publication of **DataID** metadata is released in parallel to the publishing of datasets. **DataID** was originally created under the assumption of metadata co-published alongside the datasets (**DataID** files are in the same folder or at the root of a dataset file hierarchy). This is no longer a requirement, since **DataID core** is expressive enough to find the associated dataset on the Web. Yet, storing metadata in close proximity to its described dataset is good practice and eases the discoverability of both.

In addition I propose the following best practice for publishing datasets on a given server:

Similarly to a `robot.text` file[93], I propose a file to be put in the top-level directory of the a server. The format should be Turtle, which, in my opinion, is the RDF serialisation featuring the best compromise between readability and file size. Its content is shall be a collection of relative URI paths, referencing files containing an instance of `dcat:Catalog`, pointing out datasets in turn, or instances of `dcat:CatalogRecord` (i.e. instances of `dataid:DataId`). The file should simply be called datacalaog.ttl.

In example, next to the `http://dbpedia.org/robots.txt` that explicitly excludes certain directories from being crawled, the `http://dbpedia.org/datacalaog.ttl` explicitly states where **DBpedia** datasets can be found as well as their content and relevant metadata. This best practice allows publishers to aggregate descriptions of all hosted datasets in one place and also enables users to easily discover and access these datasets. It facilitates easy aggregation of an institution's or project's datasets and massively cuts down on the time spent on navigating and searching for relevant datasets on diverse web sites.

please revise

Exploitation of DataIDs This stage summarises all activities of dataset consumers which are exploiting the benefits of the pertaining metadata as well. But exploitation is not limited to consumers alone. Data publishers, or pertaining organisations benefit from reliable and machine-readable metadata as well.

For example, the tables of the official download page for **DBpedia** releases are all based on the **DataID** documents accompanying the **DBpedia** datasets[94].

[93] `http://www.robotstxt.org`
[94] `http://wiki.dbpedia.org/downloads-2016-04`

6.3 Checklist for Publishing RDF Data

The considerable experience members of the **DBpedia** Association[95] have gathered in regard to creation and dissemination of RDF data, is distilled in a concise generalization of important steps necessary for releasing an RDF dataset. Drawing from the publishing process for **DBpedia** dataset releases, I carved out a comprehensive checklist, offering guidance when producing and releasing RDF data. While this checklist is focused on Linked Data datasets, many items present could be generalised to fit broader use cases for any type of data. I will highlight those items, which apply to dataset metadata (i.e. **DataID**).

Documentation When publishing Linked Open Data, a comprehensive documentation about data sources, ontology, versioning and other important context needs to be presented in an easy and accessible manner.

- Documentation is an iterative and never ending process. Therefore, improve documentation as early as possible to cover every aspect from the outset.
- Facilitate means to gather feedback from dataset consumers (mailing list, issue tracker, etc.).
- Record your plans on maintenance and versioning. A well maintained and regularly updated dataset will attract more consumers.

Sources Basing the generation of data on specific data sources is key for a coherent dataset release. A comprehensive documentation of the original sources should be one of the first steps in any data release.

- Record the exact origin, version and evolution of your datasources.
- Source metadata (i.e. size, location, temporal information, etc.) is also desirable
- Provide links to source files or host them yourself. Access to original data is vital for reproducibility.
- Refer to external vocabularies & ontologies utilized by your source data.
- Discuss quirks and unexpected issues you discovered about the source data.

Ontology & Mappings Deciding on an ontology for the chosen domain is deciding on conceptual entities and relationships that will represent your data-

[95] http://wiki.dbpedia.org/dbpedia-association

source. The appropriate level of abstraction is as vital as creating the mapping between both datasets. Note that all types of data extraction processes use mappings. Some mappings are obvious such as R2RML [56], while others not so obvious (i.e. hidden in software code).

- If no existing ontology can represent your data, engineer your own.
- Document any schema or mapping components.
- Before starting an extraction enrich and align your schema and mappings.
- During an extraction use a static snapshot of used ontologies and mappings. These versions are the schematical foundation of your upcoming release and are not to be changed in the extraction and publishing process.
- Use a version control system to maintain your ontology and mappings.

Software Document exactly what kind of software, in what version is used for every step of the pre-processing, extraction and post-processing cycles.

- When using custom software, use a version control system.
- Provide sufficient information about the environment you are working with.
- Provide additional information about deployment steps and configuration if necessary for the extraction process.
- Create software snapshots to enable reproducibility of an extraction process.

Extraction & Dataset Generation While the generation process of an RDF dataset is a publisher-dependent process, there are some significant points to consider.

- Prefer generating resulting datasets in different syntax formats.
- One triple per line is a preferred approach to enable easier procession of the RDF files with non-RDF tools.
- Group RDF datasets by category or context.
- Use a consistent and precise naming strategy.
- Store provenance information. Provenance information can also be achieved using the context field in quads.

Validation Validating extraction results to confirm their syntactic and semantic correctness is a necessary step every publisher needs to take.

- Try to integrate validation directly in the extraction & generation process to redude post-validation steps.
- Dataset size is an obvious indication of a release status.
- A manual inspection of triples with sampling can identify errors early in the release process.
- Statistical metrics is a good means to provide an overview about quality, especially in big RDF datasets
- Create your own validation tools and use external, general purpose RDF validation tools.
- Use a staging SPARQL endpoint and/or linked data interface for browsing the data

Enrichment Dataset enrichment with information from other data sources is a useful step to increase the precision and/or coverage of a knowledge base. However, Enrichment is a step that is closely coupled to a dataset and is therefore no specific advice in this regard.

- RDFS or OWL inferencing using the main or external schemata is definitely a means to add missing knowledge
- Dataset-dependant heuristics can be employed
- Different types of enrichment can be performed before and after the *linking* step but in most case must be done prior to validation.

Linking As a basic principle of Linked Open Data, linking is one of the most important steps in the process of creating RDF datasets. This can be achieved by multiple approaches:

- Performing a manual linking over time. (tends to be a very limited solution)
- Copying existing link sets from other datasets to previous versions of the extracted dataset (which will be outdated).
- Detecting identifiers pertaining to entities of the source or result datasets in different datasets. Generating owl:sameAs triples would complete this method.
- By using the transitive trait of owl:sameAs.
- Provide interlinks between same resources in a dataset (e.g. in different language editions)

Publishing Publishing a dataset is a complex task, which can be done in many ways. Providing raw RDF data alone is not enough to disseminate Linked Open Data.

- Update user documentation.
- Announce the release at different venues.
- Publish/update machine readable metadata (i.e. VOID, DataID, etc.) as well as online data catalogues (i.e. datahub.io).
- Upload your dump files to an accessible location with consistent serialization formats.
- If possible, make the data available through a SPARQL Endpoint and/or a Linked Data interface.

7 Application: Data Management Plans (DMP)

Over the last years Data Management Plans (DMP) have become a requirement for project proposals within most major research funding institutions. It states what types of data and metadata are employed, which limitations apply, where responsibilities lie and how the data is stored, both during research project, and after the project is completed.

The use case described here will introduce an extension to the **DataID core** ontology to extensively describe a Data Management Plan for digital data in a universal way, laying the foundation for tools helping researchers and funders with the drafting and implementing DMPs. Based on multiple requirements, raised from different DMP guidelines, I will showcase the creation of a **DataID** extension. I incorporated the re3data ontology to describe repositories and institutions, exemplifying the use of external ontologies. This will demonstrate the application of the best practices introduced in section 6.2 for the stages 'Specify' and 'Model' of the lifecycle for dataset metadata.

7.1 Specifying requirements of a DMP

The following requirements were distilled from an extensive list of DMP guidelines of different research funding bodies, covering most of the non-functional demands raised pertaining to digital datasets.

1. Describe how data will be shared (incl. repositories and access procedures).
2. Describe the procedures put in place for long-term preservation of the data.
3. Describe the types of data and metadata, as well as identifiers used.
4. Provisioning of copyright and license information, including other possible limitations to the reusability of the data.

5. Outline the rights and obligations of all parties as to their roles and responsibilities in the management and retention of research data.
6. Provision for changes in the hierarchy of involved agents and responsibilities (e.g. a Primary Investigator (PI) leaving the project).
7. Include provenance information on how datasets were used, collected or generated in the course of the project. Reference standards and methods applied.
8. Include statements on the usefulness of data for the wider public needs or possible exploitations for the likely purposes of certain parties.
9. Provide assistance for dissemination purposes of (open) data, making it easy to discover it on the web.
10. Is the metadata interoperable allowing data exchange between different meta data formats, researchers and organisations?
11. Project costs associated with implementing the DMP during and after the project. Justify the prognosticated costs.
12. Support the data management life cycle for all data produced.

Guidelines and checklists on Data Management Plans of different research (related) organisations were surveyed to generate this list of requirements. The most influential guidelines are listed below. References like **(R1)** refer to the requirements listed above and connects guidelines or checklists to a requirement respectively. A complete list of involved organisations is available on the Web[96].

1. **Horizon 2020 (H2020)**: Horizon 2020 is a framework programme of the European Commission funding research, technological development, and innovation[97]. (DMP guidelines[98]) **(R1)**,**(R2)**,**(R3)**,**(R4)**,**(R5)**,**(R8)**,**(R11)**
2. **National Science Foundation (NSF)**: The National Science Foundation[99] is a United States government agency that supports fundamental research and education in all the non-medical fields of science and engineering. (DMP guidelines[100] **(R1)**,**(R2)**),**(R4)**,**(R7)**,**(R9)**,**(R11)**,**(R12)**
3. **Economic and Social Research Council (ESRC)**: The Economic and Social Research Council[101] is one of the seven Research Councils in

[96] http://wiki.dbpedia.org/use-cases/data-management-plan-extension-dataid\#Organisation
[97] https://ec.europa.eu/programmes/horizon2020/
[98] https://ec.europa.eu/research/participants/data/ref/h2020/grants_manual/hi/oa_pilot/h2020-hi-oa-data-mgt_en.pdf
[99] http://www.nsf.gov/
[100] http://nsf.gov/eng/general/ENG_DMP_Policy.pdf
[101] http://www.esrc.ac.uk/

the United Kingdom and provides funding and support for research and training work in social and economic issues. (DMP guidelines[102])**(R3),(R4),(R8),(R11)**

4. **Deutsche Forschungsgemeinschaft (DFG)**: The DFG[103] is the largest independent research funding organisation in Germany. It promotes the advancement of science and the humanities by funding research projects, research centres and networks. (DMP guidelines[104])**(R1),(R3),(R4),(R7),(R9),(R11)**
5. **Inter-university Consortium for Political and Social Research (ICPSR)**: ICPSR[105] advances and expands social and behavioral research, acting as a global leader in data stewardship and providing rich data resources and responsive educational opportunities. (DMP guidelines[106])**(R1),(R2),(R4),(R5),(R11)**
6. **UK Data Archive**: The UK Data Archive[107] is curator of the largest collection of digital data in the social sciences and humanities in the United Kingdom. (DMP checklist[108])**(R3),(R4),(R5),(R7),(R10)**

To implement these demands in an ontology, the following observations are already evident:

1. making further use of PROV-O is necessary to deal with the extensive demands for provenance,
2. a clear specification of involved agents and their responsibilities is needed,
3. an extensive description of repositories retaining the described data is inescapable.

Our goal is to provide aid for researchers in drafting a DMP and implementing it with all requirements in mind: during the proposal phase, while the project is ongoing and the long term implementation of the DMP.

[102] http://www.esrc.ac.uk/funding/guidance-for-grant-holders/research-data-policy/
[103] http://www.dfg.de/en/
[104] http://www.dfg.de/en/research_funding/proposal_review_decision/applicants/submitting_proposal/research_data/
[105] https://www.icpsr.umich.edu/
[106] https://www.icpsr.umich.edu/icpsrweb/content/datamanagement/dmp
[107] http://www.data-archive.ac.uk/
[108] http://www.data-archive.ac.uk/create-manage/planning-for-sharing/data-management-checklist

7.2 Modelling the DataID approach

The final DMP use case extension, created to resolve the requirements listed in the last section, will extend **DataID core** together with the existing extension *Activities & Plans* and the repository ontology of *re3data.org* (cf. section 3.2.6). All three ontologies will be outlined in this section in a concise manner. For an exhaustive description please refer to the online documentation referenced below.

Activities & Plans To utilise the full breath of provenance offered by PROV-O and to add further semantics to satisfy all requirements, is the goal of this extension from the mid-layer of the **DataID Ecosystem** (cf. section 4.2). Besides PROV-O and **DataID core**, it imports the following ontologies:

DLO The Data Lifecycle Ontology[109] provides a set of conceptual entities, agents, activities, and roles to represent the general data engineering process. Furthermore, it is the basis for deriving specific domain ontologies which represent lifecycles of data engineering projects such as **DBpedia**. With the incorporation of DLO into the **DataID Ecosystem**, **DataID** extends to to meet an important requirement of the ALIGNED project, namely, to facilitate information exchange needs of combined software and (big) data engineering [9].

ORG The Organization Ontology[110] "[...] describes a core ontology for organizational structures, aimed at supporting linked data publishing of organizational information across a number of domains. It is designed to allow domain-specific extensions to add classification of organizations and roles, as well as extensions to support neighbouring information such as organizational activities." [**ReynoldsOrg**] ORG is a W3C recommendation and provides the backdrop of organisational structures, which is an important detail when describing the interplay of Agents and Activities.

put this into related work ???

mention that earlier?

To integrate DLO in the existing landscape of **DataID** and PROV-O concepts (Figure 7.1), I extended the list of sub class axioms for `dataid:Dataset`, to inherit `dlo:DataEntity`[111]. DLO is used primarily to classify different types of activities (`dlo:LifeCycleProcess`) and to adopt its basic semantics for an exchange of data artefacts between activities. Similar to `dataid:associatedAgent` and

[109] https://w3id.org/dlo

[110] https://www.w3.org/TR/vocab-org/

[111] Note: this is no ontology hijacking in a narrower sense, since one reason for creating the **DataID Ecosystem** was to provide a controlled environment for redefining concepts or properties in the extension.

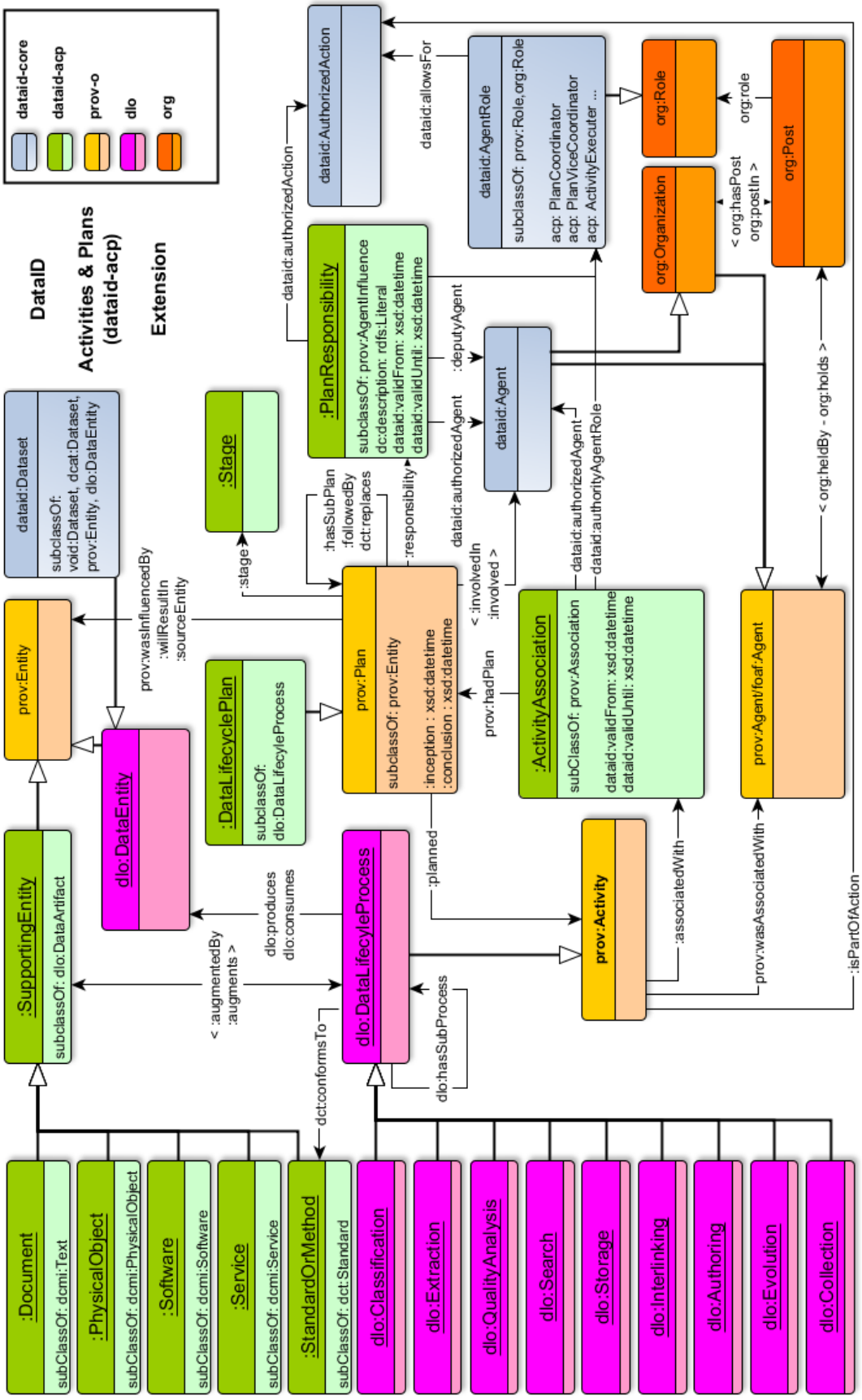


Figure 7.1: Activities & Plans extension

`dataid:Authorization` (section 5.2.6), the property `prov:wasAssociatedWith` is further qualified by the class `dataid-acp:ActivityAssociation`, reusing relevant properties for this kind of mediator (such as `dataid:authorizedAgent`). This class can not only qualify a generic association with an Agent, AgentRole and a time interval, but provides the means of referencing a `prov:Plan`, which was responsible for the scheduling of an Activity in the first place.

Plans offer the necessary properties to (pre-) define a sequence of Activities, governed by Agents (with responsibilities), which are generating Entities. An instance of `prov:Plan` is constituted of multiple sub-plans, which have a defined order. This order specifies when an Activity is executed. While `dataid-acp:ActivityAssociation` specifies responsibilities for a specific Activity, `dataid-acp:PlanResponsibility` defines Responsibilities of Agents in regard to the Plan itself (such as define order of activities etc.).

The re3data.org ontology To cover the requirements for preservation (e.g. **(R1)**), a comprehensive description of repositories is necessary. The **re3data** schema (cf. section 3.2.6) does provide a thorough description of repositories and the unique opportunity to incorporate an existing, up-to-date collection of research repositories in future **DataID**-based applications. To accomplish the integration into the DMP ontology extension, I transformed the current XML-based schema into an OWL-ontology, using established vocabularies like PROV-O and ORG. The schema as well as the data provided by **re3data** will be available as Linked Data (e.g. via **re3data** ReSTful-API), thus making it discoverable and more easily accessible for services and applications, reaching a larger circle of users. This effort was support generously by the re3data.org project.

Alongside the repository concept, a rudimentary description of institutions which are hosting or funding a repository is needed to ensure long-term sustainability and availability of a repository. The derived **re3data** ontology (Figure 7.2) supplements `r3d:Repository` and `r3d:Institution` with fitting PROV-O subclasses (`prov:Entity`, `prov:Organization`), making them subject to provenance descriptions. The ORG ontology is used to further extend the Institution class, providing organisational descriptions.

In the ontology version of the **re3data** schema, I tried to combine multiple similarly structured XML elements into on concept, where possible. For example, access regulations to the repository and the research data must be clarified, as well as the terms of use. The **re3data** ontology unifies all license and policy objects under the class `r3d:Regulation`, using the property `dct:license` to point out `odrl:Policy` descriptions of licenses, as used in the **DataID** ontology. The

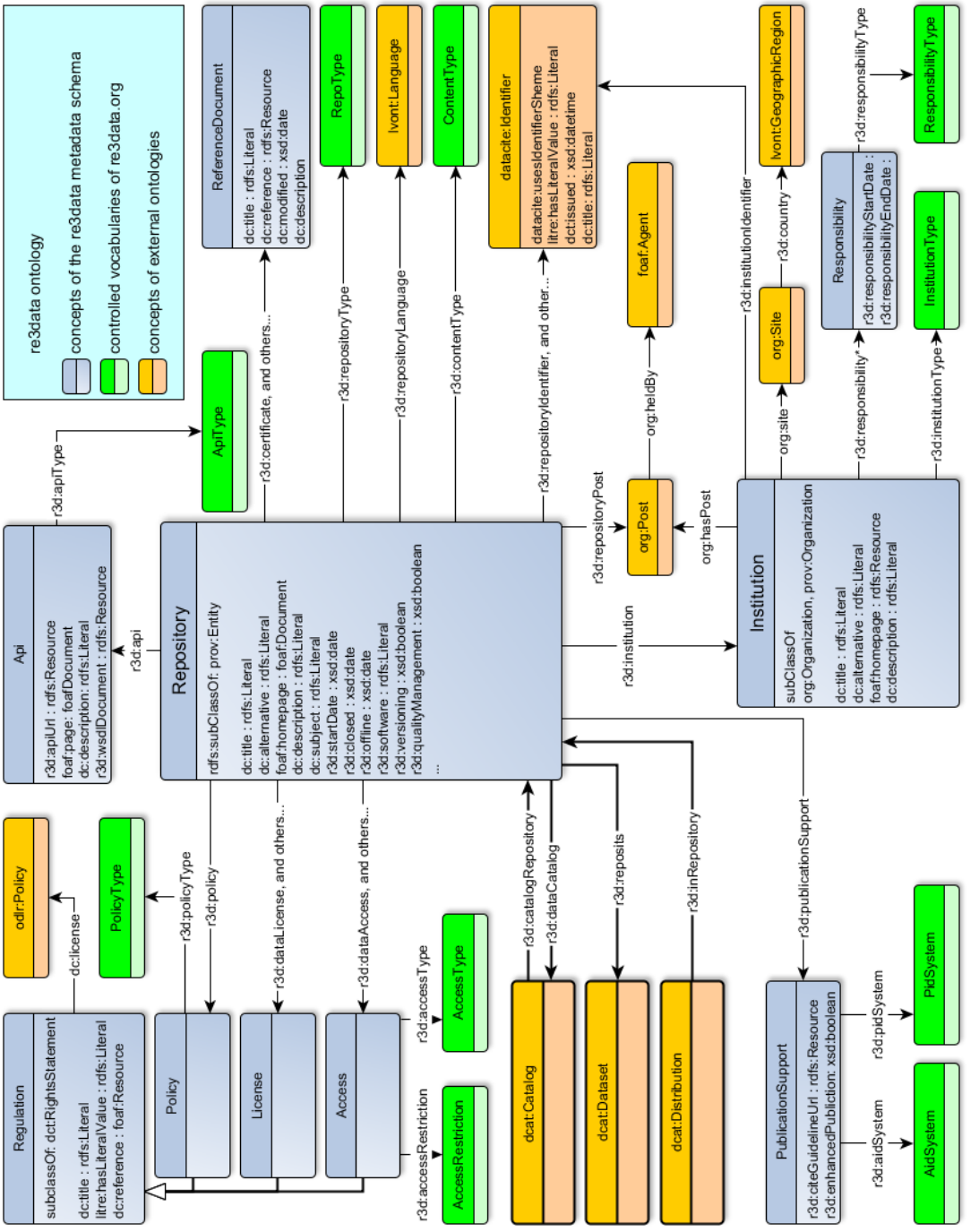


Figure 7.2: re3data.org ontology

Note: This is a reduced depiction of the ontology omitting some properties and all instances of controlled vocabularies (green boxes). A complete depiction is available here:
`https://github.com/re3data/ontology/blob/master/visuals/r3dOntology.png`
The current version can be accessed here:
`https://github.com/re3data/ontology/blob/master/r3dOntology.ttl`

ranges of multiple properties (i.e. `r3d:certificate`, `r3d:metadataStandard` and `r3d:syndication`) were bundled to form `r3d:ReferenceDocument` (a sub class of `foaf:Document`).

As with **`DataID core`**, I tried to replace commonly used concepts with existing classes of well established vocabularies. Property `r3d:repositoryLanguage` is pointing out instances of class `lvont:Language` (of the Lexvo ontology - cf. section 3.2.3) and properties calling for identifier like structures are referencing instances of `datacite:Identifier` (a natural fit, due to the fact that re3data.org is now under the care of DataCite).

By linking to `dcat:Catalog` via `r3d:dataCatalog` and `dcat:Dataset` with `r3d:reposits`, we introduced the necessary means to relate descriptions of data stored inside a repository. By providing this interface with the DCAT vocabulary, **`DataID`** can be used for the description of data in the **`re3data`** context.

On top of these foundational components, we added additional semantics, solving the requirements listed in **??** and creating the DMP extension. Extensive use of the PROV-O ontology and the concepts and properties introduced by the Activities & Plans extension is key to DMP, providing the means for describing sources and origin activities of datasets **(R7)**.

In the same vein, using the `dataid:Authorization` concept, augmented with a DMP specific set of `dataid:AgentRole` and `dataid:AuthorizedAction`, adds necessary provenance and satisfies requirement **(R5)** and **(R6)**.

A description of repositories involved in a DMP is provided by the concept `r3d:Repository`, including exact documentation of APIs and access procedures **(R1)**. More detailed information on the type of data or additional software necessary to access the data, was introduced with `dataid:Distribution`.

As in DataID core, information about licenses and other limitations are provided via `dct:license` and `dct:rights` **(R4)**, or the complementary properties of the re3data ontology concerning access and other policies. Helpful information on usefulness, reusability and other subjects for possible users of the portrayed datasets are added to the `dataid:Dataset` concept: `dataid:usefulness`, `dataid:reuseAndIntegration`, `dataid:exploitation` etc. **(R8)**.

Requirement **(R3)** is intrinsic to DataID and needs no further representation, while **(R10)** is exemplified by the next section.

Several functional requirements raised by the guidelines of research funding bodies (which are not included in the requirements of this section) will be covered by the DataID service (cf. **??**). It will provide a versioning system for DataIDs (based on properties like `dataid:nextVersion`), enabling features like tracking changes to a DataID over time. Thereby, the full data management life cycle

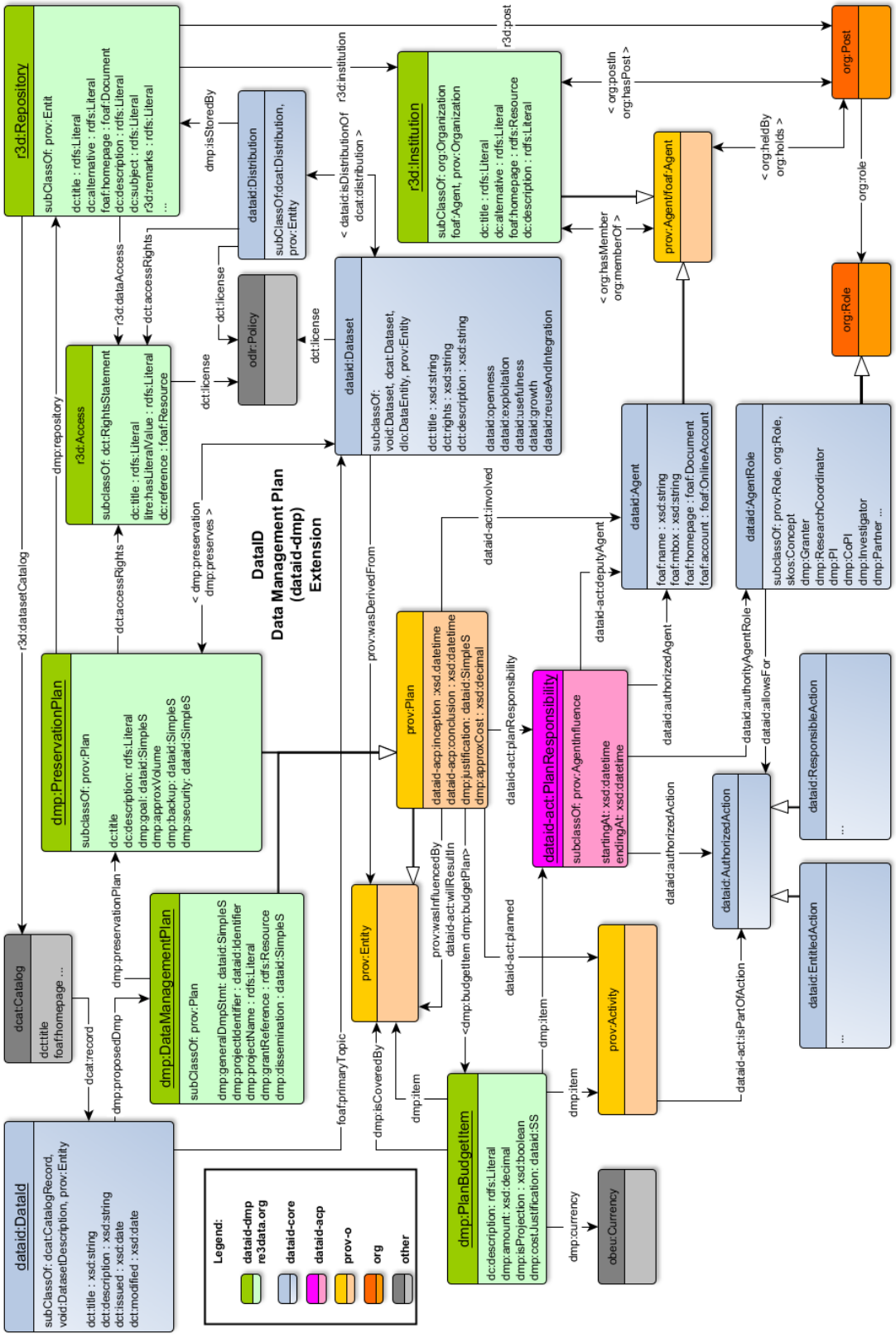


Figure 7.3: DMP use case extension

of datasets is supported **(R12)**, which spans all phases of a Data Management Plan, but this is outside of the scope of this document.

The heart of the DMP extension are two subclasses of `prov:Plan`: The `dmp:DataManagementPlan` provides the most general level of textual statements about the DMP itself or the planned dissemination process **(R9)**, as well as the necessary references to pertaining projects. While `dmp:PreservationPlan` entities can describe different approaches for preservation of different datasets **(R2)** or provide temporal scaling (e.g. regarding embargo periods). Besides textual statements about general goals and provisions for security and backup, using the `dataid-acp:planned` property to point out specific tasks, put in place to preserve data long term, is one of the more notable provenance information.

The concept `dmp:BudgetItem` is an optional tool to list costs pertaining to activities, responsibilities (consequently costs of agents) and any entity involved in a plan like `dmp:PreservationPlan`. Together with `dmp:approxCost` and `dmp:justification` it satisfies requirement **(R11)**.

As a summary; I created 3 classes and 17 properties, which, together with the concepts and properties introduced by the re3data ontology, can describe Data Management Plans as demanded by the requirements of **??**. An example of a DataID with DMP extension has been created by the ALIGNED H2020 project (e.g. the English DBpedia dataset[112]).

7.3 Evaluating of the DMP solution

[112] `http://downloads.dbpedia.org/2015-10/core-i18n/en/2015-10_dataid_en.ttl`

8 DataID and the Component MetaData Infrastructure (CMDI)

The Component MetaData Infrastructure (CMDI) is a component-based framework for the creation and utilisation of metadata schemata[**BROEDER10.163**]. It allows the distributed development of metadata components (defined as sets of related elements) and their combination to profiles in any level of detail, forming the basis for the creation of resource-specific XML Schemata and around one million publicly available metadata files. CMDI is a flexible metadata framework, which can be applied to resources from any scientific field of interest. It is especially relevant in the context of the European research infrastructure CLARIN[**Hinrichs2014**] where it is used to describe resources with a focus on the humanities and social sciences.

The very flexible and open approach of the CMDI which allows for its wide applicability, may lead in parts to problems regarding consistency and **interoperability**. Despite being rich in descriptive metadata, some CMD profiles lack consistent information of the kind stated in **??**. This includes the explicit specification of involved persons, descriptions of authoritative structures as well as technical details and actual download locations. Earlier work on the conversion of CMD profiles into RDF/RDFS[**DW2014**] reflects the complete bandwidth of CMDI-based metadata, but also some idiosyncrasies that may constrain its usage in other contexts. It is expected that a transformation of relevant data to a uniform, DataID-based vocabulary will enhance visibility and exploitation of CMDI resources in new communities. We created explicit mappings for CMD profiles, accountable for 56% of all publicly available metadata files, matching the appropriate DataID classes and applied them on all respective instance files via XSPARQL[113]. An overview of created mappings can be found on Github[114].

The creation and further adaptation of these mappings showed that the support of data considered essential in DataID differs between all profiles. The summary table 8.1 demonstrates this effect for primary properties of `dataid:Dataset` and

[113] https://www.w3.org/Submission/xsparql-language-specification/
[114] https://github.com/dbpedia/Cmdi-DataID-mappings

CMD profile	CMD instances (in % of all)	Supported properties of dataid:Dataset	Supported dataid:AgentRoles
OLAC-DcmiTerms	156.210 (17,4%)	13	3
Song	155.403 (17,3%)	9	1
imdi-session	100.423 (11,2%)	9	2
teiHeader	87.533 (9,7%)	10	2

Table 8.1: Most popular CMD profiles and their completeness regarding DataID classes

the support of different agent roles specified in `dataid:Agent`. Apparently there is a varying degree of conformance of both approaches, indicating possible shortcomings in specific CMD profiles. An example for such a potential deficit is the fine-grained modelling of involved persons or organisations via DataID's Agent concept that is only partially supported in most profiles.

9 Evaluation

10 Conclusion and Future Work

Am Ende der Arbeit steht die Zusammenfassung, die alle wichtigen Punkte und Ergebnisse der Arbeit in einfachen Worten wiedergibt. Anschließend folgt ein Ausblick auf anschließende Arbeiten und Themenvorschläge.

Das Kapitel sollte zwischen einer und drei Seiten umfassen.

add prefix table

Glossar

Vocabulary On the Semantic Web, vocabularies define the concepts and relationships (also referred to as "terms") used to describe and represent an area of concern. Vocabularies are used to classify the terms that can be used in a particular application, characterize possible relationships, and define possible constraints on using those terms. In practice, vocabularies can be very complex (with several thousands of terms) or very simple (describing one or two concepts only)[57].

Ontology There is no clear division between what is referred to as "vocabularies" (see Vocabulary) and "ontologies". The trend is to use the word "ontology" for more complex, and possibly quite formal collection of terms, whereas "vocabulary" is used when such strict formalism is not necessarily used or only in a very loose sense. Vocabularies are the basic building blocks for inference techniques on the Semantic Web[57].

Prefix Gloassar

dcat DCAT

prov Provenance Ontology

dct Dublin Core Terms

References

[1] Fadi Maali, DERI, and NUI Galway. *Data Catalog Vocabulary (DCAT)*. W3C Recommendation. URL: https://www.w3.org/TR/vocab-dcat/.

[2] Phil Archer and Keith Jeffery. *Smart Descriptions & Smarter Vocabularies Report*. W3C Workshop report. URL: https://www.w3.org/2016/11/sdsvoc/report.html.

[3] Bernadette Farias Loscio, Caroline Burle, and Newton Calegari. *Data on the Web Best Practices*. W3C Proposed Recommendation. W3C, Dec. 2016. URL: https://www.w3.org/TR/2016/PR-dwbp-20161215/.

[4] Deborah McGuinness, Timothy Lebo, and Satya Sahoo. *The PROV Ontology*. W3C Recommendation. URL: http://www.w3.org/TR/prov-o/.

[5] Mo McRoberts and Victor Rodriguez Doncel. *ODRL Version 2.1 Ontology*. W3C Community Group Specification. http://www.w3.org/ns/odrl/2/ODRL21. W3C, Mar. 2015. URL: https://www.w3.org/ns/odrl/2/.

[6] Kevin Feeney, Gavin Mendel-Gleason, and Rob Brennan. "Linked data schemata: fixing unsound foundations". In: *Semantic Web Journal-Special Issue on Quality Management of Semantic Web Assets* (2016).

[7] Mark D. Wilkinson et al. "The FAIR Guiding Principles for scientific data management and stewardship". In: *Scientific Data* 3 (Mar. 2016), pp. 160018+. ISSN: 2052-4463. DOI: 10.1038/sdata.2016.18. URL: http://dx.doi.org/10.1038/sdata.2016.18.

[8] Martin Brümmer et al. "DataID: Towards Semantically Rich Metadata for Complex Datasets". In: *Proceedings of the 10th International Conference on Semantic Systems*. SEM '14. Leipzig, Germany: ACM, 2014, pp. 84–91.

[9] Monika Solanki et al. "Enabling Combined Software and Data Engineering at Web-Scale: The ALIGNED Suite of Ontologies". In: *The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part II*. 2016, pp. 195–203. DOI: 10.1007/978-3-319-46547-0_21. URL: http://dx.doi.org/10.1007/978-3-319-46547-0_21.

[10] Markus Freudenberg et al. "The Metadata Ecosystem of DataID". In: *Metadata and Semantics Research: 10th International Conference, MTSR 2016, Göttingen, Germany, November 22-25, 2016, Proceedings*. Ed. by Emmanouel Garoufallou et al. Cham: Springer International Publishing, 2016, pp. 317–332. ISBN: 978-3-319-49157-8. DOI: 10.1007/978-3-319-49157-8_28. URL: http://dx.doi.org/10.1007/978-3-319-49157-8_28.

[11] Chaim Zins. "Conceptual Approaches for Defining Data, Information, and Knowledge: Research Articles". In: *J. Am. Soc. Inf. Sci. Technol.* 58.4 (Feb. 2007), pp. 479–493. ISSN: 1532-2882. DOI: 10.1002/asi.v58:4. URL: http://dx.doi.org/10.1002/asi.v58:4.

[12] Ramez Elmasri and Shamkant B. Navathe. *Fundamentals of Database Systems (5th Edition)*. Addison Wesley, Mar. 2006. ISBN: 0321369572. URL: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike04-20%7B%5C&%7Dpath=ASIN/0321369572.

[13] F. Bodendorf. *Daten- und Wissensmanagement*. Springer-Lehrbuch. Springer, 2003. ISBN: 9783540001027. URL: https://books.google.de/books?id=f1wjPAAACAAJ.

[14] Claude Shannon. "A Mathematical Theory of Communication". In: *Bell System Technical Journal* 27 (July 1948), pp. 379–423, 623–656. URL: http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf.

[15] Jennifer Rowley. "The Wisdom Hierarchy: Representations of the DIKW Hierarchy". In: *J. Inf. Sci.* 33.2 (Apr. 2007), pp. 163–180. ISSN: 0165-5515. DOI: 10.1177/0165551506070706. URL: http://dx.doi.org/10.1177/0165551506070706.

[16] NISO. *Understanding metadata*. ISBN 1-880124-62-9. National Information Standards Organization. 2004. URL: http://www.niso.org/standards/resources/UnderstandingMetadata.pdf.

[17] IFLA. *Functional Requirements for Bibliographic Records: Final Report*. K. G. Saur, 1998. URL: http://www.amazon.com/Functional-Requirements-Bibliographic-Records-Publications/dp/359811382X.

[18] Shane McCarron et al. *RDFa Core 1.1 - Third Edition*. W3C Recommendation. http://www.w3.org/TR/2015/REC-rdfa-core-20150317/. W3C, Mar. 2015.

[19] 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: (2001). URL: http://www.sciam.com/article.cfm?id=the-semantic-web%5C&%5C#38;print=true.

[20] Markus Lanthaler, David Wood, and Richard Cyganiak. *RDF 1.1 Concepts and Abstract Syntax*. W3C Recommendation. http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/. W3C, Feb. 2014. URL: http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/;print=true.

[21] Sebastian Rudolph et al. *OWL 2 Web Ontology Language Primer (Second Edition)*. W3C Recommendation. http://www.w3.org/TR/2012/REC-owl2-primer-20121211/. W3C, Dec. 2012. URL: https://www.w3.org/TR/owl2-primer/;print=true.

[22] Achille Fokoue et al. *OWL 2 Web Ontology Language Profiles (Second Edition)*. W3C Recommendation. http://www.w3.org/TR/2012/REC-owl2-profiles-20121211/. W3C, Dec. 2012. URL: https://www.w3.org/TR/2012/REC-owl2-profiles-20121211/;print=true.

[23] Ramanathan Guha and Dan Brickley. *RDF Schema 1.1*. W3C Recommendation. http://www.w3.org/TR/2014/REC-rdf-schema-20140225/. W3C, Feb. 2014. URL: http://www.w3.org/TR/2014/REC-rdf-schema-20140225/;print=true.

[24] Jon Barwise. "An Introduction to First-Order Logic". In: *HANDBOOK OF MATHEMATICAL LOGIC*. Elsevier BV, 1977, pp. 5–46. DOI: 10.1016/s0049-237x(08)71097-8. URL: http://dx.doi.org/10.1016/S0049-237X(08)71097-8.

[25] Ian Horrocks et al. "OWL: a Description Logic Based Ontology Language for the Semantic Web". In: *The Description Logic Handbook: Theory, Implementation, and Applications (2nd Edition)*. Ed. by Franz Baader et al. Cambridge University Press, 2007. Chap. 14. URL: download/2003/HPMW07.pdf.

[26] C. Bizer, T. Heath, and T. Berners-Lee. "Linked data - the story so far". In: *Int. J. Semantic Web Inf. Syst.* 5.3 (2009), 1?22.

[27] Tim Berners-Lee. *Linked Data*. 2006. URL: https://www.w3.org/DesignIssues/LinkedData.html (visited on 12/15/2016).

[28] Fadi Maali et al. "Enabling Interoperability of Government Data Catalogues." In: *EGOV*. Ed. by Maria Wimmer et al. LNCS. Springer, 2010.

[29] Keith G Jeffery and Anne Asserson. "Position Paper: Why CERIF?" In: 2016. URL: %5Curl%7Bhttps://www.w3.org/2016/11/sdsvoc/SDSVoc16_paper_15%7D.

[30] Keith Alexander and Michael Hausenblas. "Describing linked datasets - on the design and usage of void, the ?vocabulary of interlinked datasets". In: *In Linked Data on the Web Workshop (LDOW 09), in conjunction with 18th International World Wide Web Conference (WWW 09*. 2009.

[31] Christoph Böhm, Johannes Lorey, and Felix Naumann. "Creating voiD descriptions for Web-scale data". In: *J. Web Sem.* 9.3 (2011), pp. 339–345. DOI: 10.1016/j.websem.2011.06.001. URL: http://dx.doi.org/10.1016/j.websem.2011.06.001.

[32] Sebastian Neumaier, Jürgen Umbrich, and Axel Polleres. "Challenges of mapping current CKAN metadata to DCAT". In: *W3C Workshop on Data and Services Integration*. Amsterdam, the Netherlands, Nov. 2016. URL: https://www.w3.org/2016/11/sdsvoc/SDSVoc16_paper_16.

[33] John P. McCrae et al. "One ontology to bind them all: The META-SHARE OWL ontology for the interoperability of linguistic datasets on the Web". In: *Proc. of 12th Extended Semantic Web Conference (ESWC 2015) Satellite Events, Portorož, Slovenia*. Vol. 9341. June 2015, pp. 271–282.

[34] DCAT Application profile working group. *DCAT-AP v1.1*. 2016. URL: https://joinup.ec.europa.eu/asset/dcat_application_profile/asset_release/dcat-ap-v11 (visited on 12/15/2016).

[35] Brecht Wyns et al. "DCAT Application Profile for Data Portals in Europe". In: 2016. URL: %5Curl%7Bhttps://www.w3.org/2016/11/sdsvoc/SDSVoc16_paper_30%7D.

[36] SEMANTIC WEB HEALTH CARE and LIFE SCIENCES (HCLS) INTEREST GROUP. *HCLS mission statement*. 2016. URL: https://www.w3.org/blog/hcls/ (visited on 12/15/2016).

[37] Michel Dumontier et al. "The health care and life sciences community profile for dataset descriptions". In: *PeerJ* (2016). DOI: 10.7717/peerj.2331. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4991880/.

[38] Keith Jeffery and Anne Asserson. "CERIF-CRIS for the European e-Infrastructure". In: *Data Science Journal* 9 (2010), CRIS1–CRIS6. DOI: 10.2481/dsj.CRIS1. URL: http://dx.doi.org/10.2481/dsj.CRIS1.

[39] Luc Moreau and Paolo Missier. *PROV-DM: The PROV Data Model*. W3C Recommendation. http://www.w3.org/TR/2013/REC-prov-dm-20130430/. W3C, Apr. 2013.

[40] Luc Moreau et al. "The Rationale of PROV". In: *Web Semant.* 35.P4 (Dec. 2015), pp. 235–257. ISSN: 1570-8268. DOI: 10.1016/j.websem.2015.04.001. URL: http://dx.doi.org/10.1016/j.websem.2015.04.001.

[41] Gerard de Melo. "Lexvo.org: Language-Related Information for the Linguistic Linked Data Cloud". In: *Semantic Web* 6.4 (Aug. 2015), pp. 393–400.

[42] Dan Brickley and Libby Miller. *FOAF Vocabulary Specification 0.99*. Specification. http://xmlns.com/foaf/spec/20140114.html. Aug. 2014. URL: http://xmlns.com/foaf/spec/20140114.html.

[43] Silvio Peroni et al. "DataCite2RDF: Mapping DataCite Metadata Schema 3.1 Terms to RDF". In: (Feb. 2016). DOI: 10.6084/m9.figshare.2075356.v1. URL: https://figshare.com/articles/DataCite2RDF_Mapping_DataCite_Metadata_Schema_3_1_Terms_to_RDF/2075356.

[44] Henrik Frystyk Nielsen. *Interoperability and Evolvability*. https://www.w3.org/Protocols/Design/Interevol.html.

[45] Gregory Williams. *SPARQL 1.1 Service Description*. W3C Recommendation. http://www.w3.org/TR/2013/REC-sparql11-service-description-20130321/. W3C, Mar. 2013.

[46] Richard Cyganiak et al. *The RDF Data Cube Vocabulary*. W3C Recommendation. URL: https://www.w3.org/TR/vocab-data-cube/.

[47] *Shapes Constraint Language (SHACL)*. W3C Working Draft. https://www.w3.org/TR/2016/WD-shacl-20160814/. W3C, Aug. 2016.

[48] SPDX Workgroup a Linux Foundation Project. *SPDX Specifications*. 2016. URL: https://spdx.org/specifications (visited on 12/15/2016).

[49] Ned Freed and Nathaniel Borenstein. *RFC2046: Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*. 1996. URL: http://tools.ietf.org/html/rfc2046.

[50] Boris Villazón-Terrazas et al. "Methodological Guidelines for Publishing Government Linked Data Linking Government Data". In: *Linking Government Data*. Ed. by David Wood. New York, NY: Springer New York, 2011. Chap. 2, pp. 27–49. ISBN: 978-1-4614-1766-8. DOI: 10.1007/978-1-4614-1767-5_2. URL: http://dx.doi.org/10.1007/978-1-4614-1767-5%5C_2.

[51] Bernadette Hyland, Ghislain Auguste Atemezing, and Boris Villazón-Terrazas. *Best Practices for Publishing Linked Data*. W3C Note. http://www.w3.org/TR/2014/NOTE-ld-bp-20140109/. W3C, Jan. 2014.

[52] Claire C Austin et al. *Key components of data publishing: Using current best practices to develop a reference model for data publishing*. Dec. 2015. DOI: 10.5281/zenodo.34542. URL: https://doi.org/10.5281/zenodo.34542.

[53] Sören Auer et al. "Managing the Life-Cycle of Linked Data with the LOD2 Stack." In: *International Semantic Web Conference (2)*. Vol. 7650. Lecture Notes in Computer Science. Springer, 2012, pp. 1–16. URL: http://dblp.uni-trier.de/db/conf/semweb/iswc2012-2.html#AuerBDEHILMMNSTW12.

[54] Tomás Knap et al. "UnifiedViews: An ETL Framework for Sustainable RDF Data Processing." In: *ESWC (Satellite Events)*. Vol. 8798. Lecture Notes in Computer Science. Springer, 2014, pp. 379–383. ISBN: 978-3-319-11954-0. URL: http://dblp.uni-trier.de/db/conf/esws/eswc2014s.html#KnapKMSTV14.

[55] Dimitris Kontokostas et al. "Databugger: A Test-driven Framework for Debugging the Web of Data". In: *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion*. WWW Companion '14. Seoul, Korea: International World Wide Web Conferences Steering Committee, 2014, pp. 115–118. ISBN: 978-1-4503-2745-9. DOI: 10.1145/2567948.2577017. URL: http://jens-lehmann.org/files/2014/www_demo_databugger.pdf.

[56] Seema Sundara, Richard Cyganiak, and Souripriya Das. *R2RML: RDB to RDF Mapping Language*. W3C Recommendation. http://www.w3.org/TR/2012/REC-r2rml-20120927/. W3C, Sept. 2012.

[57] W3C. *VOCABULARIES*. https://www.w3.org/standards/semanticweb/ontology. [Online; accessed 01-December-2016]. 2015.

Appendix I

```
@prefix dataid: <http://dataid.dbpedia.org/ns/core#> .
@prefix dataid-ld: <http://dataid.dbpedia.org/ns/ld#> .
@prefix dataid-mt: <http://dataid.dbpedia.org/ns/mt#> .
@prefix dcat:  <http://www.w3.org/ns/dcat#> .
@prefix datacite: <http://purl.org/spar/datacite/> .
@prefix void:  <http://rdfs.org/ns/void#> .
@prefix spdx:  <http://spdx.org/rdf/terms#> .
@prefix owl:   <http://www.w3.org/2002/07/owl#> .
@prefix dmp:   <http://dataid.dbpedia.org/ns/dmp#> .
@prefix xsd:   <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix prov:  <http://www.w3.org/ns/prov#> .
@prefix foaf:  <http://xmlns.com/foaf/0.1/> .
@prefix dc:    <http://purl.org/dc/terms/> .
@prefix sd:    <http://www.w3.org/ns/sparql-service-description#> .

@base <http://downloads.dbpedia.org/2015-10/core-i18n/ar/2015-10_> .

<dataid_ar.ttl>
        a                           dataid:DataId ;
        dataid:associatedAgent      <http://wiki.dbpedia.org/dbpedia-association> , <http://wiki.
             dbpedia.org/dbpedia-association/persons/Freudenberg> ;
        dataid:inCatalog            <http://downloads.dbpedia.org/2015-10/2015-10_dataid_catalog.ttl>
             ;
        dataid:latestVersion        <http://downloads.dbpedia.org/2016-04/core-i18n/ar/2016-04
             _dataid_ar.ttl> ;
        dataid:nextVersion          <http://downloads.dbpedia.org/2016-04/core-i18n/ar/2016-04
             _dataid_ar.ttl> ;
        dataid:previousVersion      <http://downloads.dbpedia.org/2015-04/core-i18n/ar/2015-04
             _dataid_ar.ttl> ;
        dataid:underAuthorization   <dataid_ar.ttl?auth=maintainerAuthorization> , <dataid_ar.ttl?auth
             =creatorAuthorization> ;
        dc:hasVersion               <dataid_ar.ttl?version=1.0.0> ;
        dc:issued                   "2016-08-02"^^xsd:date ;
        dc:modified                 "2016-10-13"^^xsd:date ;
        dc:publisher                <http://wiki.dbpedia.org/dbpedia-association> ;
        dc:title                    "DataID metadata for the Arabic DBpedia"@en ;
        foaf:primaryTopic           <dataid_ar.ttl?set=maindataset> .

#### Agents & Authorizations ####

<http://wiki.dbpedia.org/dbpedia-association/persons/Freudenberg>
        a                         dataid:Agent ;
        dataid:hasAuthorization   <dataid_ar.ttl?auth=maintainerAuthorization> ;
        dataid:identifier         <http://www.researcherid.com/rid/L-2180-2016> ;
        foaf:mbox                 "freudenberg@informatik.uni-leipzig.de" ;
        foaf:name                 "Markus Freudenberg" .

<dataid_ar.ttl?auth=maintainerAuthorization>
        a                           dataid:Authorization ;
        dataid:authorityAgentRole   dataid:Maintainer ;
```

```
        dataid:authorizedAgent     <http://wiki.dbpedia.org/dbpedia-association/persons/Freudenberg>
             ;
        dataid:authorizedFor       <dataid_ar.ttl> ;
        dataid:isInheritable       true .

<http://www.researcherid.com/rid/L-2180-2016>
        a                              dataid:Identifier ;
        dataid:literal                  "L-2180-2016" ;
        dc:issued                      "2016-08-01"^^xsd:date ;
        dc:references                  <http://www.researcherid.com/rid/L-2180-2016> ;
        datacite:usesIdentifierScheme  datacite:researcherid .

<http://wiki.dbpedia.org/dbpedia-association>
        a                         dataid:Agent ;
        dataid:hasAuthorization   <dataid_ar.ttl?auth=creatorAuthorization> ;
        foaf:homepage             <http://dbpedia.org> ;
        foaf:mbox                 "dbpedia@infai.org" ;
        foaf:name                 "DBpedia Association" .

<dataid_ar.ttl?auth=creatorAuthorization>
        a                           dataid:Authorization ;
        dataid:authorityAgentRole   dataid:Creator ;
        dataid:authorizedAgent      <http://wiki.dbpedia.org/dbpedia-association> ;
        dataid:authorizedFor        <dataid_ar.ttl> ;
        dataid:isInheritable        true .

<https://wikimediafoundation.org>
        a                         dataid:Agent ;
        dataid:hasAuthorization   <http://dbpedia.org/dataset/pages_articles?lang=ar&dbpv=2016-04&file
             =pages_articles_ar.xml.bz2&auth=publisherAuthorization> , <http://dbpedia.org/dataset/
             pages_articles?lang=ar&dbpv=2016-04&auth=publisherAuthorization> ;
        foaf:mbox                 "info@wikimedia.org" ;
        foaf:name                 "Wikimedia Foundation, Inc." .

<dataid_ar.ttl?auth=publisherAuthorization>
        a                           dataid:Authorization ;
        dataid:authorityAgentRole   dataid:Creator, dataid:Publisher ;
        dataid:authorizedAgent      <https://wikimediafoundation.org> ;
        dataid:authorizedFor        <dataid_ar.ttl?set=pages_articles> ;
        dataid:isInheritable        true .

########## Main Dataset ##########

<dataid_ar.ttl?set=maindataset>
        a                        dataid:Superset ;
        dataid:associatedAgent   <http://wiki.dbpedia.org/dbpedia-association> , <http://wiki.dbpedia.
             org/dbpedia-association/persons/Freudenberg>  ;
        dataid:growth               <dataid_ar.ttl?stmt=growth> ;
        dataid:openness             <dataid_ar.ttl?stmt=openness> ;
        dataid:reuseAndIntegration  <dataid_ar.ttl?stmt=reuseAndIntegration> ;
        dataid:similarData          <dataid_ar.ttl?stmt=similarData> ;
        dataid:usefulness           <dataid_ar.ttl?stmt=usefulness> ;
        dc:description           """DBpedia is a crowd-sourced community effort to extract structured
             information from Wikipedia and make this information available on the Web. DBpedia allows
              you to ask sophisticated queries against Wikipedia, and to link the different data sets
             on the Web to Wikipedia data. We hope that this work will make it easier for the huge
             amount of information in Wikipedia to be used in some new interesting ways. Furthermore,
             it might inspire new mechanisms for navigating, linking, and improving the encyclopedia
             itself."""@en ;
        dc:hasVersion            <dataid_ar.ttl?version=1.0.0> ;
        dc:issued                "2016-07-02"^^xsd:date ;
        dc:language              <http://lexvo.org/id/iso639-3/ara> ;
        dc:license               <http://purl.oclc.org/NET/rdflicense/cc-by-sa3.0> ;
```

```
        dc:modified             "2016-08-01"^^xsd:date ;
        dc:publisher            <http://wiki.dbpedia.org/dbpedia-association> ;
        dc:rights               <dataid_ar.ttl?rights=dbpedia-rights> ;
        dc:title                "DBpedia root dataset for Arabic, version 2015-10"@en ;
        void:subset             <dataid_ar.ttl?set=long_abstracts_en_uris>, <dataid_ar.ttl?set=
             interlanguage_links> ;
        void:vocabulary         <http://downloads.dbpedia.org/2015-04/dbpedia_2015-10.owl> ;
        dcat:keyword            "maindataset"@en , "DBpedia"@en ;
        dcat:landingPage        <http://dbpedia.org/> ;
        foaf:isPrimaryTopicOf   <dataid_ar.ttl> ;
        foaf:page               <http://wiki.dbpedia.org/Downloads2015-10> .

############ Datasets #############

<dataid_ar.ttl?set=interlanguage_links>
        a                       dataid:Dataset, dataid-ld:LinkedDataDataset ;
        rdfs:label              "interlanguage links"@en ;
        dataid:associatedAgent  <http://wiki.dbpedia.org/dbpedia-association> , <http://wiki.dbpedia.
             org/dbpedia-association/persons/Freudenberg> ;
        dc:description          "Dataset linking a DBpedia resource to the same resource in other
             languages and in Wikidata."@en ;
        dc:hasVersion           <dataid_ar.ttl?version=1.0> ;
        dc:isPartOf             <dataid_ar.ttl?set=maindataset> ;
        dc:issued               "2016-07-02"^^xsd:date ;
        dc:language             <http://lexvo.org/id/iso639-3/ara> ;
        dc:license              <http://purl.oclc.org/NET/rdflicense/cc-by-sa3.0> ;
        dc:modified             "2016-08-02"^^xsd:date ;
        dc:publisher            <http://wiki.dbpedia.org/dbpedia-association> ;
        dc:title                "interlanguage links"@en ;
        void:rootResource       <dataid_ar.ttl?set=maindataset> ;
        void:triples            7480764 ;
        dcat:distribution       <dataid_ar.ttl?file=interlanguage_links_ar.ttl.bz2> , <dataid_ar.ttl?
             file=interlanguage_links_ar.tql.bz2> ;
        dcat:keyword            "DBpedia"@en , "interlanguage_links"@en ;
        dcat:landingPage        <http://dbpedia.org/> ;
        sd:defaultGraph         <http://ar.dbpedia.org> ;
        foaf:page               <http://wiki.dbpedia.org/Downloads2015-10> .

<dataid_ar.ttl?set=long_abstracts_en_uris>
        a                       dataid:Dataset, dataid-ld:LinkedDataDataset ;
        rdfs:label              "long abstracts en uris"@en ;
        dataid:associatedAgent  <http://wiki.dbpedia.org/dbpedia-association> , <http://wiki.dbpedia.
             org/dbpedia-association/persons/Freudenberg> ;
        dataid:qualifiedDatasetRelation   <dataid_ar.ttl?relation=source&target=pages_articles> ;
        dataid:relatedDataset   <dataid_ar.ttl?set=pages_articles> ;
        dc:description          "Full abstracts of Wikipedia articles, usually the first section.
             Normalized resources matching English DBpedia."@en ;
        dc:hasVersion           <dataid_ar.ttl?version=1.0.0> ;
        dc:isPartOf             <dataid_ar.ttl?set=maindataset> ;
        dc:issued               "2016-07-02"^^xsd:date ;
        dc:language             <http://lexvo.org/id/iso639-3/ara> ;
        dc:license              <http://purl.oclc.org/NET/rdflicense/cc-by-sa3.0> ;
        dc:modified             "2016-08-02"^^xsd:date ;
        dc:publisher            <http://wiki.dbpedia.org/dbpedia-association> ;
        dc:title                "long abstracts en uris"@en ;
        void:rootResource       <dataid_ar.ttl?set=maindataset> ;
        void:triples            232801 ;
        void:sparqlEndpoint      <http://dbpedia.org/sparql> ;
        dcat:distribution       <dataid_ar.ttl?sparql=DBpediaSparqlEndpoint> , <dataid_ar.ttl?file=
             long_abstracts_en_uris_ar.ttl.bz2> , <dataid_ar.ttl?file=long_abstracts_en_uris_ar.tql.
             bz2> ;
        dcat:keyword            "long_abstracts_en_uris"@en , "DBpedia"@en ;
        dcat:landingPage        <http://dbpedia.org/> ;
```

```
        sd:defaultGraph         <http://ar.dbpedia.org> ;
        foaf:page               <http://wiki.dbpedia.org/Downloads2015-10> .

<dataid_ar.ttl?set=pages_articles>
        a                           dataid:Dataset ;
        rdfs:label                  "Wikipedia XML source dump file"@en ;
        dataid:associatedAgent      <https://wikimediafoundation.org> ;
        dataid:needsSpecialAuthorization  <dataid_ar.ttl?auth=publisherAuthorization> ;
        dc:description              "The Wikipedia dump file, which is the source for all other
             extracted datasets."@en ;
        dc:hasVersion               "20160305" ;
        dc:issued                   "2016-03-05"^^xsd:date ;
        dc:language                 <http://lexvo.org/id/iso639-3/ara> ;
        dc:license                  <http://purl.oclc.org/NET/rdflicense/cc-by-sa3.0> ;
        dc:publisher                <https://wikimediafoundation.org> ;
        dc:title                    "Wikipedia XML source dump file"@en ;
        dcat:distribution           <http://dbpedia.org/dataset/pages_articles?lang=ar&dbpv=2016-04&
             file=pages_articles_ar.xml.bz2> ;
        dcat:keyword                "Wikipedia"@en , "XML dump file"@en ;
        dcat:landingPage            <https://meta.wikimedia.org/wiki/Data_dumps> .

########## Distributions ###########

<dataid_ar.ttl?file=interlanguage_links_ar.ttl.bz2>
        a                           dataid:SingleFile ;
        rdfs:label                  "interlanguage_links_ar.ttl.bz2" ;
        dataid:associatedAgent      <http://wiki.dbpedia.org/dbpedia-association> , <http://wiki.
             dbpedia.org/dbpedia-association/persons/Freudenberg> ;
        dataid:checksum             <dataid_ar.ttl?file=interlanguage_links_ar.ttl.bz2&checksum=md5>
              ;
        dataid:isDistributionOf     <dataid_ar.ttl?set=interlanguage_links> ;
        dataid:preview              <http://downloads.dbpedia.org/preview.php?file=2015-10_sl_core-
             i18n_sl_ar_sl_interlanguage_links_ar.ttl.bz2> ;
        dataid:uncompressedByteSize  1184687767 ;
        dc:description              "Dataset linking a DBpedia resource to the same resource in
             other languages and in Wikidata."@en ;
        dc:hasVersion               <dataid_ar.ttl?version=1.0> ;
        dc:issued                  "2016-07-02"^^xsd:date ;
        dc:license                  <http://purl.oclc.org/NET/rdflicense/cc-by-sa3.0> ;
        dc:modified                 "2016-08-02"^^xsd:date ;
        dc:publisher                <http://wiki.dbpedia.org/dbpedia-association> ;
        dc:title                    "interlanguage links"@en ;
        dcat:byteSize               62761863 ;
        dcat:downloadURL            <http://downloads.dbpedia.org/2015-10/core-i18n/core-i18n/ar/
             interlanguage_links_ar.ttl.bz2> ;
        dcat:mediaType              dataid-mt:MediaType_turtle_x-bzip2 .

<dataid_ar.ttl?file=interlanguage_links_ar.tql.bz2>
        a                           dataid:SingleFile ;
        rdfs:label                  "interlanguage_links_ar.tql.bz2" ;
        dataid:associatedAgent      <http://wiki.dbpedia.org/dbpedia-association> , <http://wiki.
             dbpedia.org/dbpedia-association/persons/Freudenberg> ;
        dataid:checksum             <dataid_ar.ttl?file=interlanguage_links_ar.tql.bz2&checksum=md5>
              ;
        dataid:isDistributionOf     <dataid_ar.ttl?set=interlanguage_links> ;
        dataid:preview              <http://downloads.dbpedia.org/preview.php?file=2015-10_sl_core-
             i18n_sl_ar_sl_interlanguage_links_ar.tql.bz2> ;
        dataid:uncompressedByteSize  1598056873 ;
        dc:description              "Dataset linking a DBpedia resource to the same resource in
             other languages and in Wikidata."@en ;
        dc:hasVersion               <dataid_ar.ttl?version=1.0> ;
        dc:issued                  "2016-07-02"^^xsd:date ;
        dc:license                  <http://purl.oclc.org/NET/rdflicense/cc-by-sa3.0> ;
```

```
        dc:modified                 "2016-08-02"^^xsd:date ;
        dc:publisher                <http://wiki.dbpedia.org/dbpedia-association> ;
        dc:title                    "interlanguage links"@en ;
        dcat:byteSize               71946917 ;
        dcat:downloadURL            <http://downloads.dbpedia.org/2015-10/core-i18n/core-i18n/ar/
             interlanguage_links_ar.tql.bz2> ;
        dcat:mediaType              dataid-mt:MediaType_n-quads_x-bzip2 .

<dataid_ar.ttl?file=long_abstracts_en_uris_ar.ttl.bz2>
        a                           dataid:SingleFile ;
        rdfs:label                  "long_abstracts_en_uris_ar.ttl.bz2" ;
        dataid:associatedAgent      <http://wiki.dbpedia.org/dbpedia-association> , <http://wiki.
             dbpedia.org/dbpedia-association/persons/Freudenberg> ;
        dataid:checksum             <dataid_ar.ttl?file=long_abstracts_en_uris_ar.ttl.bz2&checksum=
             md5> ;
        dataid:isDistributionOf     <dataid_ar.ttl?set=long_abstracts_en_uris> ;
        dataid:preview              <http://downloads.dbpedia.org/preview.php?file=2015-10_sl_core-
             i18n_sl_ar_sl_long_abstracts_en_uris_ar.ttl.bz2> ;
        dataid:uncompressedByteSize  186573907 ;
        dc:description              "Full abstracts of Wikipedia articles, usually the first section
             . Normalized resources matching English DBpedia."@en ;
        dc:hasVersion               <dataid_ar.ttl?version=1.0> ;
        dc:issued                  "2016-07-02"^^xsd:date ;
        dc:license                  <http://purl.oclc.org/NET/rdflicense/cc-by-sa3.0> ;
        dc:modified                 "2016-08-02"^^xsd:date ;
        dc:publisher                <http://wiki.dbpedia.org/dbpedia-association> ;
        dc:title                    "long abstracts en uris"@en ;
        dcat:byteSize               33428372 ;
        dcat:downloadURL            <http://downloads.dbpedia.org/2015-10/core-i18n/core-i18n/ar/
             long_abstracts_en_uris_ar.ttl.bz2> ;
        dcat:mediaType              dataid-mt:MediaType_turtle_x-bzip2 .

<dataid_ar.ttl?file=long_abstracts_en_uris_ar.tql.bz2>
        a                           dataid:SingleFile ;
        rdfs:label                  "long_abstracts_en_uris_ar.tql.bz2" ;
        dataid:associatedAgent      <http://wiki.dbpedia.org/dbpedia-association> , <http://wiki.
             dbpedia.org/dbpedia-association/persons/Freudenberg> ;
        dataid:checksum             <dataid_ar.ttl?file=long_abstracts_en_uris_ar.tql.bz2&checksum=
             md5> ;
        dataid:isDistributionOf     <dataid_ar.ttl?set=long_abstracts_en_uris> ;
        dataid:preview              <http://downloads.dbpedia.org/preview.php?file=2015-10_sl_core-
             i18n_sl_ar_sl_long_abstracts_en_uris_ar.tql.bz2> ;
        dataid:uncompressedByteSize  204174726 ;
        dc:description              "Full abstracts of Wikipedia articles, usually the first section
             . Normalized resources matching English DBpedia."@en ;
        dc:hasVersion               <dataid_ar.ttl?version=1.0> ;
        dc:issued                  "2016-07-02"^^xsd:date ;
        dc:license                  <http://purl.oclc.org/NET/rdflicense/cc-by-sa3.0> ;
        dc:modified                 "2016-08-02"^^xsd:date ;
        dc:publisher                <http://wiki.dbpedia.org/dbpedia-association> ;
        dc:title                    "long abstracts en uris"@en ;
        dcat:byteSize               36026709 ;
        dcat:downloadURL            <http://downloads.dbpedia.org/2015-10/core-i18n/core-i18n/ar/
             long_abstracts_en_uris_ar.tql.bz2> ;
        dcat:mediaType              dataid-mt:MediaType_n-quads_x-bzip2 .

<dataid_ar.ttl?sparql=DBpediaSparqlEndpoint>
        a                          dataid-ld:SparqlEndpoint ;
        rdfs:label                 "The official DBpedia sparql endpoint"@en ;
        dataid:accessProcedure     <dataid_ar.ttl?stmt=sparqlaccproc> ;
        dataid:associatedAgent     <http://support.openlinksw.com/> ;
        dataid:isDistributionOf    <dataid_ar.ttl?set=long_abstracts_en_uris> ;
```

```
        dc:description           "The official sparql endpoint of DBpedia, hosted graciously by
             OpenLink Software (http://virtuoso.openlinksw.com/), containing all datasets of the /core
              directory."@en ;
        dc:hasVersion            <dataid_ar.ttl?version=1.0> ;
        dc:issued                "2016-07-02"^^xsd:date ;
        dc:license               <http://purl.oclc.org/NET/rdflicense/cc-by-sa3.0> ;
        dc:modified              "2016-08-02"^^xsd:date ;
        dc:title                 "The official DBpedia sparql endpoint"@en ;
        sd:endpoint                              <http://dbpedia.org/sparql> ;
        sd:supportedLanguage     sd:SPARQL11Query ;
        sd:resultFormat                  <http://www.w3.org/ns/formats/RDF_XML>, <http://www.w3.org/
             ns/formats/Turtle> ;
        dcat:accessURL           <http://dbpedia.org/sparql> ;
        dcat:mediaType           <http://dataid.dbpedia.org/ns/mt#MediaType_sparql-results+xml> .

########## Relations ############

<dataid_ar.ttl?relation=source&target=pages_articles>
        a                           dataid:DatasetRelationship ;
        dataid:datasetRelationRole  dataid:SourceRole ;
        dataid:qualifiedRelationOf  <dataid_ar.ttl?set=long_abstracts_en_uris> ;
        dataid:qualifiedRelationTo  <dataid_ar.ttl?set=pages_articles> .

########## Checksums ############

<dataid_ar.ttl?file=interlanguage_links_ar.ttl.bz2&checksum=md5>
        a                   spdx:Checksum ;
        spdx:algorithm      spdx:checksumAlgorithm_md5 ;
        spdx:checksumValue  "b1a6885fba528b08c53b0ad800a94f7a"^^xsd:hexBinary .


<dataid_ar.ttl?file=interlanguage_links_ar.tql.bz2&checksum=md5>
        a                   spdx:Checksum ;
        spdx:algorithm      spdx:checksumAlgorithm_md5 ;
        spdx:checksumValue  "d34de153e77570f118b7425e5cf1ca0b"^^xsd:hexBinary .

<dataid_ar.ttl?file=long_abstracts_en_uris_ar.ttl.bz2&checksum=md5>
        a                   spdx:Checksum ;
        spdx:algorithm      spdx:checksumAlgorithm_md5 ;
        spdx:checksumValue  "2503179cd96452d33becd1e974d6a163"^^xsd:hexBinary .

<dataid_ar.ttl?file=long_abstracts_en_uris_ar.tql.bz2&checksum=md5>
        a                   spdx:Checksum ;
        spdx:algorithm      spdx:checksumAlgorithm_md5 ;
        spdx:checksumValue  "ffdf034c2477d81b5aaeced0312984d4"^^xsd:hexBinary .

########## Statements ###########

<dataid_ar.ttl?rights=dbpedia-rights>
        a                 dataid:SimpleStatement ;
        dataid:literal  """DBpedia is derived from Wikipedia and is distributed under the same
             licensing terms as Wikipedia itself. As Wikipedia has moved to dual-licensing, we also
             dual-license DBpedia starting with release 3.4. Data comprising DBpedia release 3.4 and
             subsequent releases is licensed under the terms of the Creative Commons Attribution-
             ShareAlike 3.0 license and the GNU Free Documentation License. Data comprising DBpedia
             releases up to and including release 3.3 is licensed only under the terms of the GNU Free
              Documentation License."""@en .

<dataid_ar.ttl?version=1.0.0>
        a                 dataid:SimpleStatement ;
        dataid:literal  "1.0.0" .

<dataid_ar.ttl?stmt=sparqlaccproc>
```

```
        a                dataid:SimpleStatement ;
        dc:references    <https://www.w3.org/TR/sparql11-overview/> ;
        dataid:literal  "An endpoint for sparql queries: provide valid queries." .

<dataid_ar.ttl?stmt=openness>
        a                dataid:SimpleStatement ;
        dataid:statement  "DBpedia is an open dataset, licensed under CC-BY-SA 3.0."@en .

<dataid_ar.ttl?stmt=growth>
        a                dataid:SimpleStatement ;
        dataid:statement  "DBpedia is an ongoing open-source project. Goal of the project is the
             extraction of the Wikipedia, as complete as possible. Currently, 126 languages are being
             extracted. In the future, DBpedia will try to increase its importance as the center of
             the LOD cloud by adding further external datasets"@en .

<dataid_ar.ttl?stmt=similarData>
        a                dataid:SimpleStatement ;
        dataid:statement  "Similar data can be found in datasets like Freebase (https://freebase.com)
             , Wikidata (https://www.wikidata.org), Yago (http://www.mpi-inf.mpg.de/departments/
             databases-and-information-systems/research/yago-naga/yago//) or OpenCyc (http://opencyc.
             org)."@en .

<dataid_ar.ttl?stmt=usefulness>
        a                dataid:SimpleStatement ;
        dataid:statement  "DBpedia is a useful resource for interlinking general datasets with
             encyclopedic knowledge. Users profitting from DBpedia are open data developers, SMEs and
             researchers in data science and NLP"@en .

<dataid_ar.ttl?stmt=reuseAndIntegration>
        a                dataid:SimpleStatement ;
        dataid:statement  "DBpedia data can be integrated into other datasets and reused for data
             enrichment or mashup purposes"@en .


########### MediaTypes ###########

<http://dataid.dbpedia.org/ns/mt#MediaType_sparql-results+xml>
        a                   dataid:MediaType ;
        dataid:typeTemplate  "application/sparql-results+xml" ;
        dc:conformsTo        <http://dataid.dbpedia.org/ns/core> .

dataid-mt:MediaType_turtle_x-bzip2
        a                      dataid:MediaType ;
        dataid:innerMediaType  dataid:MediaType_turtle ;
        dataid:typeExtension   ".bz2" ;
        dataid:typeTemplate    "application/x-bzip2" ;
        dc:conformsTo          <http://dataid.dbpedia.org/ns/core> .

dataid-mt:MediaType_n-quads_x-bzip2
        a                      dataid:MediaType ;
        dataid:innerMediaType  dataid:MediaType_n-quads ;
        dataid:typeExtension   ".bz2" ;
        dataid:typeTemplate    "application/x-bzip2" ;
        dc:conformsTo          <http://dataid.dbpedia.org/ns/core> .

dataid:MediaType_n-quads
        a                    dataid:MediaType ;
        dataid:typeExtension  ".nq", ".tql" ;
        dataid:typeTemplate   "application/n-quads" ;
        dc:conformsTo         <http://dataid.dbpedia.org/ns/core> .

dataid:MediaType_turtle
        a                    dataid:MediaType ;
```

```
        dataid:typeExtension  ".ttl" ;
        dataid:typeTemplate   "text/turtle" ;
        dc:conformsTo         <http://dataid.dbpedia.org/ns/core> .
```

Erklärung

"'Ich versichere, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe, insbesondere sind wörtliche oder sinngemäße Zitate als solche gekennzeichnet. Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschlusses führen kann"'.

Ort Datum Unterschrift