Introducing the Data Quality Vocabulary (DQV)

Riccardo Albertoni[a,*], Antoine Isaac[b]
[a]*Istituto di Matematica Applicata e Tecnologie Informatiche "Enrico Magenes", Consiglio Nazionale delle Ricerche (IMATI-CNR), Via De Marini, 6, 16149 Genova, Italy, E-mail: albertoni@ge.imati.cnr.it*
[b]*VU University Amsterdam and Europeana, The Netherlands, E-mail: aisaac@few.vu.nl*

Abstract. The Data Quality Vocabulary (DQV) provides a metadata model for expressing data quality. DQV was developed by the Data on the Web Best Practice (DWBP) working group of the World Wide Web Consortium (W3C) between 2013 and 2017. This paper aims at providing a deeper understanding of DQV. It introduces its key design principles, main components, and the main discussion points that have been raised in the process of designing it. The paper compares DQV with previous quality documentation vocabularies and demonstrates the early uptake of DQV by collecting tools, papers, projects that have exploited and extended DQV.

Keywords: Data Quality, W3C, Metadata, RDF vocabulary, DCAT.

1. Introduction

Data quality is a well-known issue accompanying information systems in every evolution from the database systems to the current Web of Data. As discussed in the recent W3C Recommendation Data on the Web Best Practices [10], "The quality of a dataset can have a big impact on the quality of applications that use it. As a consequence, the inclusion of data quality information in data publishing and consumption pipelines is of primary importance. Documenting data quality significantly eases the process of dataset selection, increasing the chances of reuse. Independently from domain-specific peculiarities, the quality of data should be documented and known quality issues should be explicitly stated in metadata."

Aiming to facilitate the publication of such data quality information on the Web, especially in the growing area of data catalogues, the W3C Data on the Web Best Practices Working (DWBP) group has developed the Data Quality Vocabulary (DQV) [2]. DQV is a (meta)data model implemented as an RDF vocabulary, which extends the Data Catalog Vocabulary (DCAT) [24] with properties and classes suitable for expressing the quality of datasets and their distributions. DQV has been conceived as a high-level, interoperable framework that must accommodate various views over data quality. DQV does not seek to determine what "quality" means. Quality lies in the eye of the beholder: there is no objective, ideal definition of it. Some datasets will be judged as low-quality resources by some data consumers, while they will perfectly fit others' needs. There are heuristics designed to fit specific assessment situations that rely on quality indicators, such as pieces of data content, meta-information and human ratings in order to give indications about the suitability of data for some intended use. DQV re-uses the notions of quality dimensions, categories and metrics to let its users represent various approaches to data quality assessments. It also stresses the importance of allowing different actors to assess the quality of datasets and publish their annotations, certificates, or mere opinions about a dataset.

We claim that DQV exhibits by design a set of characteristics that have not been combined so far in quality documentation vocabularies, e.g., the Dataset Quality Ontology (daQ) [13,14], the Data Quality Management Vocabulary (DQM) [19], the Quality Model Ontology (QMO) [30] and the Evaluation Result ontology (EVAL) [31]: (1) it

results from a community effort; (2) it directly re-uses standard W3C vocabularies; (3) it covers a wide range of quality requirements; (4) it embraces the minimal ontological commitment. Especially, though DQV has been originally conceived to document DCAT datasets and distributions, it can be used to document the quality of any resource published on the web. DQV can then serve as common exchange ground between quality assessments from different parties as well as a building block to model specific quality assessments in a large spectrum of domains and applications.

This paper complements the published W3C Working Group Note [2], offering insight into the requirements and the process considered developing DQV. Section 2 explains our methodology, especially detailing the design principles adopted for the development of DQV; Section 3 presents the main components of DQV and illustrates how these components can represent the most common quality information; Section 4 compares DQV with related work; Section 5 discusses the current DQV uptake; Section 6 summarizes the contributions and outlines future activities.

2. Methodology and Design Principles

DQV has been developed under the umbrella of the W3C Data on the Web Best Practices (DWBP) working group, which was chartered to facilitate the development of open data ecosystems, guiding publishers and fostering the trust in the data among developers. The group worked between December 2013 and January 2017; the group discussions took place in about 135 near-weekly teleconferences and five face-to-face meetings. The group has delivered a set of best practices collected in the Data on the Web Best Practices W3C Recommendation [10] and two W3C Working Group Notes describing the RDF vocabularies: the Dataset Usage Vocabulary [35] and the Data Quality Vocabulary [2]. The efforts of the working group have focused on meeting requirements expressed in another W3C Working Group Note, the Data on the Web Best Practices Use Cases & Requirements [23].

This paper focuses on DQV. The design of DQV considers the requirements distilled in Section 4.2 of the DWBP Use Cases & Requirements [23] and the feedback received in response to four DQV Public Working Drafts issued towards relevant external communities. Public feedback and interactions about DQV with group members are registered in 90 public mailing list messages, in more than 30 formal issues, and over 130 formal and informal actions[1]. All teleconferences and meetings followed the W3C process, which generates URIs for each meeting agenda, issue, action as well as mailing list post discussed by the group. This paper explicitly refers to requirements and technical design issues, in order to lead interested readers into richly interlinked working group resources which deepen the discussion and ground the design choices made. In order to avoid systematic use of URIs, the references to group resources are made as follows:

• *Issues*. Details of all issues are documented in the Working Group's issue tracker at https://www.w3.org/2013/dwbp/track/issues/. Issues are cited in the text by number, e.g., Issue 204 for https://www.w3.org/2013/dwbp/track/issues/204.

• *Requirements*. Requirements are documented in the Use Cases & Requirements document [23]. Requirements are referred to in the text by their handles, e.g., R-QualityOpinions.

In terms of guiding principles, the group has considered two fundamental principles to enable the reusability and the uptake of DQV:

- a commitment to find a sweet spot between existing proposals rather than surpass them in scope or complexity;
- a focus on interoperability. DQV should be easy to map to (for existing vocabularies) as well as to re-use and extend.

These enabling principles turned into design principles that others might have failed to follow, and which mirror two best practices that have been identified in our Working Group's more general recommendations on data vocabularies [10]:

- minimize ontological commitment, fitting Best Practice 16 ("Choose the right formalization level");

[1] Here we count both formal actions within the W3C process, which are assigned to group members in order to address an issue and are tracked by the W3C facilities (see https://www.w3.org/2013/dwbp/track/actions/closed) and the more informal actions from the editor's to-do-list (see https://www.w3.org/2013/dwbp/wiki/Data_quality_draft_actions)

- re-use existing vocabularies unless there's a good reason not to do so (Best Practice 15).

The principles above have deeply impacted the design of DQV. For example, DQV is designed to fit well into the DCAT model, but, in compliance with the minimal ontological commitment principle, it is also possible to deploy it with other models. Consequently, no formal restrictions have been imposed to restrict the domain of DQV properties to DCAT datasets and distributions. We also decided not to define the DQV elements as part of the DCAT namespace (Issue 179). Similarly, DQV draws inspiration from the Dataset Quality Ontology (daQ) [13], but it deliberately chooses to downplay some of daQ assumptions. In particular, daQ defines Metric, Dimension, and Category as abstract classes and it imposes specific cardinality constraints on the properties that can relate them. Discussions in the working group have acknowledged the abstractness of Dimension and Category, but defining each dimension and category as a class of individuals seems not optimal in terms of representational complexity and interoperability. The group thinks there are no fundamental features of daQ that are lost in DQV representing dimensions and categories as instances of skos:Concept (as suggested for Issue 205), which also expresses that they are abstract entities. So the group has left out the use of (abstract) classes (Issue 204). The discussions in DWBP have also pointed out that the cardinality constraints adopted in daQ with respect to the attachment of metrics to dimensions (and dimensions to categories) might not apply in a wider application context where classifications are not always crisp, and some quality metrics could be classified in several dimensions. So no cardinality constraints are formally imposed on these properties (Issue 187).

In adherence to the second design principle, DQV reuses standard W3C vocabularies. In particular, it reuses SKOS [7,26] to organize the Quality Dimensions and Categories into hierarchies and to represent their lexical representations and definitions (Issue 205). It employs RDF Data Cube [12] to model the values returned by quality assessments and PROV-O [33] to model provenance and quality derivations. It also exploits some of the vocabularies that at the time were under development in other working groups: the Web Annotation Vocabulary [11] is used to model Feedback and Quality certificates; the Open Digital Rights Language (ODRL) [39] is considered to model quality policies; SHACL[2] [21] is suggested to express data integrity constraints. ShEx[3], which was not clearly available as an option at the time of writing the DQV specification, may also be used for this purpose. Conformance to standards is modeled by reusing dct:conformsTo from the DCMI Metadata Terms (see section 3.5). In total, the namespace maintained by W3C[4] specifically for DQV defines only ten new classes, nine properties and two instances.

Finally, as presented in the introduction, DQV is a data model implemented as an RDF vocabulary. The vocabulary is represented as an RDFS/OWL ontology that is accessible via the DQV namespace. This is only one of many possible representations, however. In particular, future extensions or profiles may benefit from the availability of representations of DQV axioms as SHACL or ShEx shapes.

3. DQV Components

This section describes the components of DQV. DQV relates (DCAT) datasets and distributions with different types of quality statements, which include Quality Annotations, Standards, Quality Policies, Quality Measurements and Quality Provenance. Quality information pertains to one or more quality characteristics relevant to the consumer (aka, Quality Dimensions). The way DQV represents the quality dimensions and each kind of quality statements is shown in a separate gray box in Fig. 1 and discussed in the following sections[5].

3.1. Quality Dimensions and Categories

Data quality is commonly conceived as a multi-dimensional construct [41] where each dimension represents a quality-related characteristic relevant to the consumer (e.g., accuracy, timeliness,

[2] https://www.w3.org/TR/shacl/
[3] http://shex.io/shex-semantics/
[4] http://www.w3.org/ns/dqv#
[5] Examples included in the following sections are available as RDF files at https://w3id.org/quality/DQV/examples.

completeness, relevancy, objectivity, believability, understandability, consistency, conciseness). For this reason, DQV relates Quality Metrics, Quality Annotations, Standards, and Quality Policies to Quality Dimensions (see the dqv:inDimension property in Fig. 1).

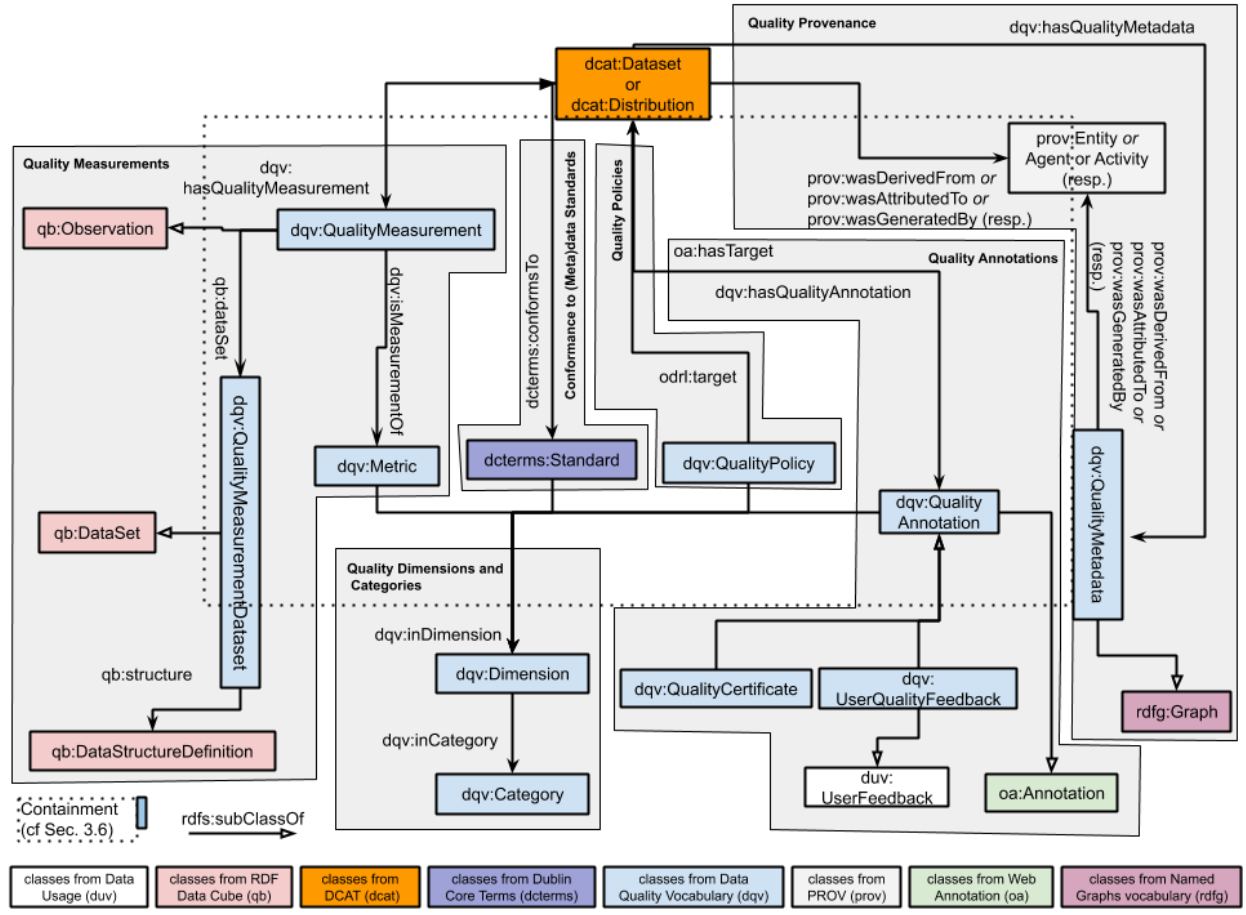

Fig. 1. Diagram depicting DQV classes and properties. For the sake of readability, the diagram does not include all the DQV properties.

The quality dimensions are systematically organized in groups referred to as quality categories. For instance, categories can be defined according to the type of information that is considered, e.g., Content-Based – based on information content itself; Context-Based – information about the context in which information was claimed; Rating-Based – based on ratings about the data itself or the information provider. But they can also be defined according to other criteria, which can lead to quite composite hierarchies depending on the idea of fitness for use that guides specific quality assessments.

In coherence with the principle of reusing existing vocabularies, DQV uses SKOS [26] to define dimensions and categories (Issue 205).

The classes dqv:Dimension and dqv:Category represent quality dimensions and categories respectively, and are defined as subclasses of skos:Concept.

Dimensions are linked to categories using the property dqv:inCategory. Distinct quality frameworks might have different perspectives over dimensions and their grouping in categories, so in accordance to the minimal ontological commitment, no specific cardinality constraints are imposed on the dqv:inCategory property.

The properties skos:prefLabel and skos:definition indicate the name and definition for dimensions and categories. SKOS semantic relations (i.e., skos:related, skos:broader, skos:narrower) are used to relate dimensions/categories. In particular,

skos:broader and skos:narrower enable to model fine-grained granularities for dimensions and categories (Issue 225). SKOS mapping relations, such as skos:exactMatch, skos:relatedMatch and skos:broaderMatch, can be used to map the dimensions and categories from independently produced classifications.

Example 1 shows a fragment, in the RDF Turtle syntax[6], of quality dimensions and categories defined according to Zaveri et al. [41][7]. It introduces two dimensions, ldqd:availability and ldqd:completeness, and the categories they belong to, ldqd:accessibilityDimensions and ldqd:intrinsicDimensions. The example also relates the defined dimensions with other dimensions among those discussed in Zaveri et al. [41].

Example 1:

```
# Definition of dimensions
ldqd:availability
        a dqv:Dimension ;
        dqv:inCategory ldqd:accessibilityDimensions ;
        skos:prefLabel "Availability"@en ;
        skos:definition "Availability of a dataset is the
extent to which data (or some portion of it) is present,
obtainable and ready for use."@en .

ldqd:completeness
        a dqv:Dimension ;
        dqv:inCategory ldqd:intrinsicDimensions ;
        skos:prefLabel "Completeness"@en;
        skos:definition "Completeness refers to the
degree to which all required information is present in a
particular dataset."@en .

# Definition of categories
ldqd:accessibilityDimensions a dqv:Category ;
        skos:prefLabel "Accessibility"@en .

ldqd:intrinsicDimensions a dqv:Category ;
        skos:prefLabel "Intrinsic dimensions"@en .

# Relations between dimensions
ldqd:completeness skos:related ldqd:conciseness,
ldqd:semanticAccuracy .
```

DQV mints only one instance of quality dimension — dqv:precision — in order to tackle the R-GranularityLevels requirement for expressing the level of detail (granularity) of a dataset, for which we could not find a dimension in existing quality frameworks. It does not define a normative list of quality dimensions. Starting from use cases included in the Use Cases & Requirements document [23], it offers as two possible starting points the quality dimensions proposed in ISO 25012 [20] and Zaveri et al. [41] (Issue 200). Ultimately, implementers will need to choose themselves the collection of quality dimensions that best fits their needs. They can extend on these starting points, creating their own refinements of categories and dimensions, and of course their own metrics. They can mix existing approaches: the DQV Working Group Note shows for example that the proposals from ISO 25012 and Zaveri et al. are not completely disjoint [2]. Implementers can also adopt completely different classifications if the existing ones do not fit their specific application scenarios[8]. They should, however, be aware that relying on existing classifications and metrics increases interoperability, i.e., the chance that human and machine agents can properly understand and exploit their quality assessments.

3.2. Quality Measurements

Quality measurements provide quantitative or qualitative information about data. Each measurement results from the application of a metric, which is a standard procedure for measuring a data quality dimension by observing concrete features in the data.

The need to represent quality measurements and metrics emerged from the use case analysis by the DWBP group and it is indicated as the R-QualityMetrics requirement: “Data should be associated with a set of documented, objective and, if available, standardized quality metrics. This set of quality metrics may include user-defined or domain-specific metrics”. Multiple metrics might refer to the same dimensions. For example, Zaveri et al. [41] discuss that the availability dimension can be evaluated using metrics based on the accessibility of a SPARQL endpoint or of an RDF dump. Typically, the measured value of a metric is numeric

[6] In this paper, examples shows constructs in bold when they are especially relevant for the feature being illustrated.

[7] We provide a non normative RDF representation of these dimensions and categories under the W3C umbrella at https://www.w3.org/2016/05/ldqd.

[8] For example, qSKOS — a quite popular tool assessing the quality of thesauri — detects a set of SKOS quality issues, which is distinct from the dimensions proposed by ISO 25012 and Zaveri et al.. To represent the results of qSKOS in DQV, we have mapped the qSKOS quality issues into a new classification of quality dimensions and categories published at http://w3id.org/quality/qskos.

(e.g., for the metric "human-readable labeling of classes, properties and entities", the percentage of entities having an rdfs:label or rdfs:comment) or boolean (e.g., whether or not a SPARQL endpoint is accessible).

DQV represents quality measurements as instances of the dqv:QualityMeasurement class. Each measurement refers through the property dqv:isMeasurementOf to a metric which is represented as an instance of the dqv:Metric class.

dqv:QualityMeasurement encodes the metric's observed value using the property dqv:value. The expected data type for dqv:value is represented at the metric level, using the property dqv:expectedDataType, so that implementers are encouraged to represent all measurements of a metric using the same data type. The unit of measure of dqv:value is expressed using the property sdmx-attribute:unitMeasure that is already used by RDF Data Cube (see below). The dqv:computedOn property refers to the resource on which the quality measurement is performed. In the DQV context, this property is generally expected to have instances of dcat:Dataset or dcat:Distribution as objects. However, in compliance with the minimal ontological commitment principle, dqv:computedOn can refer to any kind of rdfs:Resource (e.g., a dataset, a linkset, a graph, a set of triples).

Example 2 below describes three metrics, :populationCompletenessMetric, :sparqlAvailabilityMetric and :downloadURLAvailabilityMetric, which evaluate the two quality dimensions ldqd:completeness and ldqd:availability defined in Example 1. It also shows three quality measurements :measure1, :measure2 and :measure3 that represent the result of applying the above metrics to the DCAT dataset :myDataset and two distributions of it: CSV (:myCSVDatasetDistribution) and the output of a SPARQL endpoint in one this endpoint's possible output formats (:mySPARQLDatasetDistribution).

Example 2:

```
# Definition of a Dataset and its Distribution
:myDataset
   a dcat:Dataset ;
   dcterms:title "My dataset"@en ;
   dcat:distribution :myCSVDatasetDistribution ,
            :mySPARQLDatasetDistribution .

:mySPARQLDatasetDistribution
   a dcat:Distribution ;
   dcat:accessURL <http://www.example.org/sparql> ;
   dcterms:title "SPARQL access to the dataset"@en ;
   dcat:mediaType "application/sparql-results+json" .

:myCSVDatasetDistribution
   a dcat:Distribution ;
   dcat:downloadURL
            <http://www.example.org/files/mydataset.csv> ;
   dcterms:title "CSV distribution of dataset"@en ;
   dcat:mediaType "text/csv" ;
   dcat:byteSize "87120"^^xsd:decimal .

#Definition of Metrics
:populationCompletenessMetric
   a dqv:Metric ;
   skos:definition "Ratio between the number of objects
represented in the dataset and the number of objects
expected to be represented according to the declared
dataset scope."@en ;
   dqv:inDimension ldqd:completeness ;
   dqv:expectedDataType xsd:decimal .

:sparqlAvailabilityMetric
   a dqv:Metric ;
   skos:definition "It checks if a void:sparqlEndpoint is
specified for a distribution and if the server responds to a
SPARQL query."@en ;
   dqv:inDimension ldqd:availability ;
   dqv:expectedDataType xsd:boolean.

:downloadURLAvailabilityMetric
   a dqv:Metric ;
   skos:definition "Checks if dcat:downloadURL is
available and if its value is dereferenceable."@en ;
   dqv:inDimension ldqd:availability ;
   dqv:expectedDataType xsd:boolean .

#Actual metric values
:mySPARQLDatasetDistribution
    dqv:hasQualityMeasurement :measurement1 .

:myDataset dqv:hasQualityMeasurement :measurement2
.

:myCSVDatasetDistribution dqv:hasQualityMeasurement
            :measurement3 .

:measurement1
   a dqv:QualityMeasurement ;
   dqv:computedOn :mySPARQLDatasetDistribution ;
   dqv:isMeasurementOf :SPARQLAvailabilityMetric ;
   dqv:value "true"^^xsd:boolean .

:measurement2
   a dqv:QualityMeasurement ;
   dqv:computedOn :myDataset ;
   dqv:isMeasurementOf :populationCompletenessMetric;
   sdmx-attribute:unitMeasure
<http://www.wurvoc.org/vocabularies/om-1.8/Percentage> ;
   dqv:value "90.0"^^xsd:decimal .
```

```
:measurement3
   a dqv:QualityMeasurement ;
   dqv:computedOn :myCSVDatasetDistribution ;
   dqv:isMeasurementOf :downloadURLAvailabilityMetric;
   dqv:value "false"^^xsd:boolean .
```

The use of metrics checking for completeness is one of the possible approaches to indicate that data is partially missing or that a dataset is incomplete, as demanded by the R-DataMissingIncomplete and R-QualityCompleteness requirements. The systematic adoption of shared dimensions and metrics makes the quality assessments among different datasets more comparable as requested by the R-QualityComparable requirement.

Metrics can have parameters. For example, the LusTRE project has defined a metric to evaluate the quality of a set of links between a dataset and another, from the perspective of data augmentation scenarios [3]. This metric can be applied considering a specific property in the data or values that are in a specific language, in order to produce an indicator tailored to applications that relies more heavily on this property or this language (see the discussion on Data Cube and parameters below). DQV does not propose a standard representation of such parameters. The Working Group observed that parameters for metrics were a much less mature aspect of our field, and as a consequence, the DQV Group Note only suggests possible approaches, on which the users might build on their solutions (see Issue 223 and DQV Appendix D, "Defining and using parameters for metrics" [2]).

Note that in general DQV is also agnostic about the technology adopted to implement the metrics; it does not provide any specific "language for defining metrics". For example, DQV does not specify how the rule from Example 10 ("A dataset is available if at least one of its distributions is available") should be represented and evaluated.

For the definition of quality metrics and measurements, DQV has adapted and revised the ontology for Dataset Quality Information (daQ) [14]. It keeps most of the daQ structure. However, daQ vocabulary is not a community standard and its guarantee of sustainability may be judged not sufficient (Issue 180). DQV thus coins its own classes and properties, and declares equivalence statements (owl:equivalentClass or owl:equivalentProperty) with their daQ counterparts. Following the discussion in Issues 182, 186 and 231, the DWBP group revised the names of classes and properties with the aim of making more understandable what each class and property means. In particular, daq:Observation has been renamed as dqv:QualityMeasurement, daq:metric as dqv:isMeasurementOf, daq:QualityGraph as dqv:QualityMeasurementDataset.

Like daQ, DQV reuses the RDF Data Cube vocabulary [12] to represent multi-dimensional data, including statistics (Issue 191). It defines dqv:QualityMeasurement as a subclass of qb:Observation; dqv:isMeasurementOf and dqv:computedOn as instances of qb:DimensionProperty. Sets of dqv:QualityMeasurement sharing the same qb:DataStructureDefinition can be grouped in instances of dqv:QualityMeasurementDataset, which is a subclass of qb:DataSet. The reuse of RDF Data Cube maintains some of the specific advantages offered by daQ [13], for example, the quality measurements can be visualized reusing Data Cube enabled applications such as CubeViz[9], and observations can be grouped together automatically according to quality metrics, dimensions, and categories. The example below shows a Data Cube Structure that can be associated with quality measures.

Example 3:

```
# Associating measurements to a Quality Measurement
Dataset
:measurement1 qb:dataSet :linksetQualityMeasurements .
:measurement2 qb:dataSet :linksetQualityMeasurements .

# Defining the Quality Measurement Dataset
:linksetQualityMeasurements a
            dqv:QualityMeasurementDataset ;
            qb:structure  :dsd .

#Definition of a Data Cube structure
:dsd a qb:DataStructureDefinition ;
## Expressing Data Cube dimensions
    qb:component [
            qb:dimension  dqv:isMeasurementOf ;
            qb:order      1
            ] ;
    qb:component [
```

[9] http://cubeviz.aksw.org/

```
            qb:dimension  dqv:computedOn ;
            qb:order      2
           ] ;
    qb:component [
           qb:dimension  dcterms:date ;
           qb:order      3
           ] ;
## Expressing the Data Cube measure
   qb:component [ qb:measure  dqv:value ; ] ;
## Expressing the Data Cube attribute (here, unit of
measurement)
    qb:component [
           qb:attribute sdmx-attribute:unitMeasure ;
           qb:componentRequired false ;
           qb:componentAttachment qb:DataSet
           ] .
```

DQV users should be aware that applying Data Cube Data Structure Definitions to their quality statement datasets has a broad impact on the possible content of these. In fact, all the resources that are said to be in a quality measurement dataset (using the qb:dataSet property) are indeed expected to feature all the components defined as mandatory in the Data Structure Definition associated with the dataset. Moreover, RDF Cube imposes specific integrity constraints, for example, "no two qb:Observations in the same qb:DataSet may have the same value for all dimensions" [12]. Considering the Data Structure Definition in Example 3, the above constraint implies that it is not allowed to have two distinct measurements for the same metric, resource, and date. As a result, metrics depending on parameters shall be used with extra care so as to adhere to this constraint: data publishers will be able to represent quality measurements for the same metric, resource, and date, but they will need to include in the structure the distinct parameters that are applied. For example, if the metric depends on two extra parameters, such as :onProperty and :onLanguage (reprising the example of [3] mentioned above), the qb:DataStructure will include two qb:component in addition to those in Example 3.

Example 3 bis:

```
## Adding a new type to the parameter properties
   :onLanguage a qb:DimensionProperty .
   :onProperty a qb:DimensionProperty .

## Extending the structure of :dsd with two new dimensions
   :dsd qb:component [
           qb:dimension :onProperty ;
                qb:order     4
               ] ;
        qb:component [
           qb:dimension :onLanguage ;
                qb:order     5
               ] .
```

All the measurements represented in a dqv:QualityMeasurementDataset conforming to such an extended structure have to indicate metric, resource, date and the extra two parameters. Data Cube's Data Structures are also harder to apply when quality metrics relying on different parameters are mixed together.

3.3. Quality Annotations

Quality annotations include ratings, quality certificates and quality feedback that can be associated with data. DQV tackles these kinds of quality statements to meet the R-QualityOpinions and R-UsageFeedback requirements, respectively "Subjective quality opinions on the data should be supported" and "Data consumers should have a way of sharing feedback and rating data."

In accordance with the principle of re-using established vocabularies, DQV models annotations by specializing the Web Annotation Vocabulary [11]. Quality annotations are defined as instances of the dqv:QualityAnnotation class, which is a subclass of oa:Annotation (Issue 185). The dqv:UserQualityFeedback and dqv:QualityCertificate classes specialize dqv:QualityAnnotation to represent feedback that users provide on the quality of data, and certificates that guarantee the quality of the data according to a set of quality assessment rules.

In the W3C Web Annotation data model, all annotations should be provided with a motivation or purpose, using the property oa:motivatedBy in combination with instances of the class oa:Motivation (itself a subclass of skos:Concept) . For all quality annotations, the oa:motivatedBy must have as value the individual dqv:qualityAssessment defined by DQV for representing the motivation of assessing quality. Besides dqv:qualityAssessment, one of the instances of oa:Motivation predefined by the Web Annotation vocabulary should be indicated as motivation in order to distinguish among the different kinds of feedback, e.g., classifications, comments or questions (Issue 201). In accordance with the Web Annotation vocabulary, DQV uses oa:hasTarget to connect an instance of

dqv:QualityAnnotation or its subclasses (dqv:QualityCertificate and dqv:UserQualityFeedback) to the resource the annotation is about. Any kind of resource (e.g., a dataset, a linkset, a graph, a set of triples) could be a target. However, in the DQV context, this property is generally expected to be used in statements in which objects are instances of dcat:Dataset or dcat:Distribution. The oa:hasBody property is used to connect an instance of dqv:QualityAnnotation or its subclasses to the body of the annotation, e.g., a certificate or a textual comment. The property dqv:inDimension can also be used to relate instances of dqv:QualityAnnotation with quality dimension instances of dqv:Dimension.

The example below shows how to model a question about the completeness of the "City of Raleigh Open Government Data" dataset identified by the Open Data Institute (ODI) with the URI https://certificates.theodi.org/en/datasets/393. The annotation :questionQA is a user (quality) feedback, which is associated to the dataset and has as body the question, as represented in :textBody. It specifies that the user intends to ask a question about the dataset, by indicating oa:questioning as motivation.

Example 4:

```
# Expressing a question about dataset quality
<https://certificates.theodi.org/en/datasets/393> a
        dcat:Dataset ;
    dqv:hasQualityAnnotation :questionQA .

:questionQA
  a dqv:UserQualityFeedback ;
  oa:hasTarget
        <https://certificates.theodi.org/en/datasets/393> ;
  oa:hasBody :textBody ;
  oa:motivatedBy
        dqv:qualityAssessment, oa:questioning ;
  dqv:inDimension ldqd:completeness .

:textBody a oa:TextualBody ;
  rdf:value "Could you please provide information about
the completeness of your dataset?" ;
  dc:language "en" ;
  dc:format "text/plain"  .
```

Example 5 expresses that the "City of Raleigh Open Government Data" dataset is classified as a four stars dataset in the 5 Stars Linked Open Data rating system. The annotation :classificationQA is a user feedback that associates the dataset with the :four_stars concept where we expect the open data 5 stars classification to be represented through five instances of skos:Concept expressing the different ratings in an :OpenData5Star SKOS concept scheme. The feedback is a form of classification for the dataset, which is expressed by the oa:classifying motivation.

Example 5:

```
 #Expressing that a dataset fits in a quality classification

<https://certificates.theodi.org/en/datasets/393> a
dcat:Dataset ;
     dqv:hasQualityAnnotation :classificationQA .

:classificationQA
  a dqv:UserQualityFeedback ;
  oa:hasTarget
        <https://certificates.theodi.org/en/datasets/393> ;
  oa:hasBody :four_stars ;
  oa:motivatedBy
        dqv:qualityAssessment, oa:classifying ;
  dqv:inDimension ldqd:availability .

:four_stars
  a skos:Concept;
  skos:inScheme :OpenData5Star ;
  skos:prefLabel "Four stars"@en ;
  skos:definition "Dataset available on the Web with
structured machine-readable non proprietary format. It uses
URIs to denote things."@en .
```

Example 6 expresses that an ODI certificate for the "City of Raleigh Open Government Data" dataset is available at a specific URL. :myDatasetQA is an annotation connecting the dataset to its quality certificate.

Example 6:

```
# Expressing that a dataset received an ODI certificate
  <https://certificates.theodi.org/en/datasets/393> a
dcat:Dataset ;
  dqv:hasQualityAnnotation :myDatasetQA .

:myDatasetQA
  a dqv:QualityCertificate ;
  oa:hasTarget
<https://certificates.theodi.org/en/datasets/393> ;
   oa:hasBody
<https://certificates.theodi.org/en/datasets/393/certificate> ;
  oa:motivatedBy dqv:qualityAssessment  .
```

DQV users can exploit quality annotations jointly with quality metrics. For example, automatic quality checkers can complement their metric-based measurements with annotations to provide information not directly expressible as metrics values (e.g., listing errors and inconsistencies found

assessing the quality metrics). Quality annotations can be also deployed when quality metrics and measurement have not been explicitly applied, for example to describe a known completeness issue of a certain dataset.

Note that annotations can be used in conjunction with or as an alternative to other DQV components, such as metrics. The model is flexible and the choice to use an annotation instead of a metric might depend on the application context and the user preferences. As a rule of thumb, annotations seem a good fit for manual quality evaluations, while metrics and measurements would rather represent automatic assessments. But there are situations in which metrics can be measured manually, and annotations can be generated automatically. For example, a (basic) availability evaluation may be represented as a score of 0 or 1, but still set by a human evaluator; or it could be set as an annotation with two concepts ("available" and "not available") decided by an automatic agent trying to fetch the dataset at a provided URI.

3.4. Quality Policies

Quality policies are agreements between service providers and consumers that are chiefly defined by data quality concerns.

The DWBP working group decided to express such quality policies following a discussion about Service Level Agreements (SLA) (Issue 184) and suggestions received in one of the feedback rounds (Issue 202). DQV introduces the class dqv:QualityPolicy to express that a dataset follows a quality policy or agreement. DQV does not provide a complete framework for expressing policies. The class dqv:QualityPolicy is rather meant as an anchor point, through which DQV implementers can relate properties and classes of policy-dedicated vocabularies (such as ODRL [39]) to the core DQV patterns.

Example 7 below specifies that a data provider grants permission to access the dataset :myDataset of Example 2. It also commits to serve the data with a certain quality, more concretely, 99% availability in the SPARQL endpoint (seen as a DCAT distribution) :myDatasetSparqlDistribution. Such a policy is expressed in ODRL and DQV as an offer assigning to the service provider a duty on the service provider, which is expressed as a constraint on the measurement of a quality metric (:sparqlEndpointUptime). In ODRL the odrl:assigner is the issuer of the policy statement; in our case, it is also the assignee of the duty to deliver the distribution as the policy requires it. There is no recipient for the policy itself: this example is about a generic data access policy. Such assignees are likely to be found for instances of dqv:QualityPolicy that are also instances of the ODRL class odrl:Agreement.

Example 7:

```
:serviceProvider a odrl:Party .

:myDataset a  dcat:Dataset ;
  dcat:distribution :myDatasetSparqlDistribution .

:myDatasetSparqlDistribution a dcat:Distribution .

:policy1 a odrl:Offer, dqv:QualityPolicy ;
  dqv:inDimension ldqd:availability ;
  odrl:permission [
    a odrl:Permission ;
    odrl:target :myDataset ;
    odrl:action odrl:read ;
    odrl:assigner :serviceProvider;
    odrl:duty [
      a odrl:Duty;
      odrl:assignee :serviceProvider;
      odrl:target :myDatasetSparqlDistribution ;
      odrl:constraint [
        a odrl:Constraint ;
        prov:wasDerivedFrom :sparqlEndpointUptime;
        odrl:leftOperand odrl:percentage ;
        odrl:operator odrl:gteq ;
        odrl:rightOperand "99"^^xsd:double ;
      ]
    ]
  ]
```

The above example slightly differs from the example originally included in the DQV Working Group Note [2]: the ODRL vocabulary has evolved since DQV was published and the expression of ODRL constraints now requires the representation of left and right operands.

3.5. Conformance to (Meta)data Standards

Conformance to a given standard can convey crucial information about the quality of a data catalog. In particular, the requirement to model that a dataset's metadata is compliant with a standard

came out as a cross-cutting requirement discussing the relation of DQV with other standards (Issue 202) and the relation between certificate, policies, and standards in Issue 184 and Issue 199.

DQV models this kind of statement by reusing the property dcterms:conformsTo and the class dcterms:Standard. This simple solution is copied from GeoDCAT-AP [18], an extension of the DCAT vocabulary [24] conceived to represent metadata for geospatial data portals. GeoDCAT-AP allows one to express that a dataset's metadata conforms to an existing standard, following the recommendations of ISO 19115, ISO 19157 and the EU INSPIRE directive. The newly published DCAT 2 [6] has copied this pattern, too.

The DQV Working Group Note [2] includes an example to illustrate how a DCAT catalog record can be said to be conformant with the GeoDCAT-AP standard itself.

Conformance to standards can of course be also asserted for datasets themselves, not only metadata about them. The following example shows how a dataset can be declared conformant to the ISO 8601 standard, using the same basic pattern.

Example 8:

```
:myDataset a dcat:Dataset ;
   dcterms:conformsTo :ISO8601 .

:ISO8601 a dcterms:Standard;
   dcterms:title "Date and time format - ISO 8601" ;
   dcterms:comment "ISO 8601 can be used by anyone
who wants to use a standardized way of presenting dates
and times. It helps cut out the uncertainty and confusion
when communicating internationally."@en;
   dcterms:issued "2004-12-23"^^xsd:date ;
   foaf:page
<https://www.iso.org/iso-8601-date-and-time-format.html> .
```

Finer-grained representation of conformance statements can be found in the literature. Applications with more complex requirements, such as being able to represent 'non-conformance' as tested by specific procedures, may implement them. The GeoDCAT Application Profile, for example, suggests a "provisional mapping" for extended profiles, which re-uses the PROV data model for provenance (see Annex II.14 at [18]). Such solutions come however at the cost of having to publish and exchange representations that are much more elaborate. At the time we considered them, it also appeared they would have to be aligned with the result of other (then ongoing) efforts on data validation and reporting thereof, for example, in the SHACL context. The group therefore decided to postpone addressing such detailed conformance matters (see Issue 202 for more details).

3.6. Quality Provenance

The DWBP WG has identified a requirement for tracking provenance for metadata in general (R-ProvAvailable). Quality statements expressed in DQV qualify as metadata and DQV tracks the provenance of quality statements by reusing W3C's Provenance Ontology [33]. DQV specifically introduces the dqv:QualityMetadata class to group and "reify" quality-related statements into graphs, which can be used to represent the provenance of these statements using PROV-O properties. DQV especially foresees the use of the properties prov:wasDerivedFrom, prov:attributedTo, and prov:wasGeneratedBy.

QualityMetadata containers can contain every kind of quality statements supported in DQV. However, they do not necessarily have to include all types of quality statements. Implementers define the granularity of containment as they see fit. For example, they might want to group together the results from the same tools, the same type of quality statements, or all quality statements from the same quality assessment campaign. In the current version, DQV leaves also open the choice of the technical means used for containment. Implementers can use (RDF) graph containment[10] to assign quality statements to specific graphs, for example using RDF TriG[11]. As an alternative, they can also use an appropriate property — for example (a subproperty of) dcterms:hasPart — to link instances of dqv:QualityMetadata with instances of other DQV classes (Issue 181). It would also be possible to use RDF's standard statement reification approach[12], linking instances of dqv:QualityMetadata to the instances of rdf:Statement that constitute that metadata. This would however subject the data to the well-known issues of RDF's reification (especially, the fact that a reified statement does not imply the original statement, which therefore needs to be also stated in a "regular" way). Finally, this

[10] https://www.w3.org/TR/rdf11-primer/#section-multiple-graphs

[11] https://www.w3.org/TR/trig/

[12] https://www.w3.org/TR/rdf-schema/#ch_reificationvocab

sort of containment can also be expressed in non-RDF knowledge representation models. For example the Wikibase data model[13] used by Wikidata would allow one to relate a quality statement to an item that stands for a DQV dataset, by using a dedicated *qualifier*.

The example below gathers a set of quality statements on :myDataset and its distributions :mySPARQLDatasetDistribution, :myCSVDatasetDistribution including measurements (:measurement1, :measurement2 and :measurement3) and an annotation (:classificationOfmyDataset) produced during the same activity (:myQualityChecking) by the tool (:myQualityChecker).

Example 9

```
# linking dataset and distribution to the quality metadata
:myDataset dqv:hasQualityMetadata :myQualityMetadata.

:mySPARQLDatasetDistribution dqv:hasQualityMetadata
            :myQualityMetadata .

:myCSVDatasetDistribution dqv:hasQualityMetadata
            :myQualityMetadata.

# :myQualityMatadata is a graph expressed according to
TRIG syntax (see https://www.w3.org/TR/trig/)
 :myQualityMetadata {

 :measurement1
   a dqv:QualityMeasurement ;
   dqv:computedOn :mySPARQLDatasetDistribution ;
   dqv:isMeasurementOf :SPARQLAvailabilityMetric ;
   dqv:value "true"^^xsd:boolean .

 :measurement2
   a dqv:QualityMeasurement ;
   dqv:computedOn :myDataset ;
   dqv:isMeasurementOf :populationCompletenessMetric;
   sdmx-attribute:unitMeasure
<http://www.wurvoc.org/vocabularies/om-1.8/Percentage> ;
   dqv:value "90.0"^^xsd:decimal .

 :measurement3
   a dqv:QualityMeasurement ;
   dqv:computedOn :myCSVDatasetDistribution ;
   dqv:isMeasurementOf :downloadURLAvailabilityMetric ;
   dqv:value "false"^^xsd:boolean .

 :classificationOfmyDataset
   a dqv:UserQualityFeedback ;
   oa:hasTarget  :myDataset ;
   oa:hasBody :four_stars ;
   oa:motivatedBy
            dqv:qualityAssessment, oa:classifying ;
   dqv:inDimension ldqd:availability .
}

# :myQualityMetadata has been created by
:myQualityChecker and it is the result of the
:myQualityChecking activity

:myQualityMetadata
   a dqv:QualityMetadata ;
   prov:wasAttributedTo :myQualityChecker ;
   prov:generatedAtTime
"2015-05-27T02:52:02Z"^^xsd:dateTime ;
   prov:wasGeneratedBy :myQualityChecking .

# :myQualityChecker is a service computing some quality
metrics

:myQualityChecker
   a prov:SoftwareAgent ;
   rdfs:label "A quality assessment service"^^xsd:string .

# Further details about quality service/software can be
provided, for example, deploying vocabularies such as
Dataset Usage Vocabulary (DUV), Dublin Core or Asset
Description Metadata Schema for Software (ADMS.SW)

# :myQualityChecking is the activity that has generated
:myQualityMetadata from :myDatasetDistribution

:myQualityChecking
   a prov:Activity;
   rdfs:label "The checking of :myDataset and its
distributions quality"^^xsd:string;
   prov:wasAssociatedWith :myQualityChecker;
   prov:used   :myDataset,
:mySPARQLDatasetDistribution,
:myCSVDatasetDistribution ;
   prov:generated  :myQualityMetadata;
   prov:startedAtTime
"2015-05-27T00:52:02Z"^^xsd:dateTime;
   prov:endedAtTime
"2015-05-27T02:55:00Z"^^xsd:dateTime .
```

At a lower level of granularity, DQV allows to track provenance links across quality measurements or annotations. It is possible to use PROV-O's prov:wasDerivedFrom to indicate that a quality statement, say, a certificate, is derived from another, for example, the computation of some metric (Issue 222). At a higher level of abstraction, DQV foresees that more abstract quality constructs such as Metrics, Standards and Policies can also be explicitly derived one from another. For example, the availability of a dataset can be defined in terms of the availability of its distributions. Depending on the application, a dataset can be considered available if each or at least one of its distributions are available. The metric defined in the example below

[13] https://www.mediawiki.org/wiki/Wikibase/DataModel/Primer

assumes a dataset is available if at least one of its distributions are available. The example shows how DQV models such derivation at the level of metrics (as well as of the corresponding measurements) by using the property prov:wasDerivedFrom.

Example 10:

```
:datasetAvailabilityMetric
   a dqv:Metric ;
    prov:wasDerivedFrom :downloadURLAvailabilityMetric,
:SPARQLAvailabilityMetric;
     skos:definition "Checks the availability of the specified
distributions. A dataset is available if at least one of its
distributions is available."@en ;
   dqv:expectedDataType xsd:boolean ;
   dqv:inDimension  ldqd:availability .

:measurement4
   a dqv:QualityMeasurement ;
   dqv:computedOn :myDataset ;
   dqv:isMeasurementOf :datasetAvailabilityMetric ;
   prov:wasDerivedFrom
            :measurement1, :measurement3 ;
   dqv:value "false"^^xsd:boolean .
```

Note that DQV does not systematically declare its classes to be subclasses of those in the Provenance Ontology (e.g., subclassing between dqv:QualityMeasurement and prov:Entity). First, "recognizing" that some DQV resources have also a PROV-O type can be inferred by the (RDFS) domain or range of the PROV-O properties applied to them. Second, we wanted to avoid limiting in any way the use of PROV-O with DQV, as well as the proliferation of instances declaring PROV-O classes without any other actual provenance statements associated to them.

4. Prior Work on Expressing Quality Metadata

The linked data community has proposed different quality documentation vocabularies in the last eight years.

The Data Quality Management Vocabulary (DQM) [19] addresses the definition of quality problems for representing quality rules and data cleansing, and defines more than 40 brand new classes and 56 properties embedding most common quality problems. It is an early work which seems not being maintained anymore. It explicitly models data quality dimensions such as Accuracy, Completeness and Timeliness, but dimensions are hard coded in the model rather than expressed as a quality dimension framework that can be plugged in. DQM models the notion of quality score that relates to quality concepts such as Quality Metric and Measurements, but it does not include other DQV quality components such as annotations and policies.

The Quality Model Ontology (QMO) [30] and Evaluation Result ontology (EVAL) [31] are two ontologies defined to work together; QMO defines "a generic ontology for representing quality models and their resources" [29]; EVAL defines "a generic ontology for representing results obtained in an evaluation process" [29]. QMO and EVAL are not developed by an international working group, but they explicitly adopt the terminology used by the ISO 25010 (SQuaRE) and by the ISO/IEC 15939 standards. Similarly to DQV, they represent metrics and measurement results. However, they do not include other DQV quality components such as annotations and policies, nor do they reuse the W3C vocabularies such as SKOS to represent quality metrics and dimensions.

The Dataset Quality Ontology (daQ) "allows for representing data quality metrics as statistical observations in multi-dimensional spaces" [13]. DQV borrows quality metrics and dimensions from daQ, but it revises the daQ solution according to the minimal ontological commitment and the reuse of best-of-breed W3C vocabularies. Besides, DQV covers quality components such as Quality feedback, certificate and policy that are not included in daQ. In the family of the ontologies around daQ, the Quality Problem Report Ontology QPRO supports the representation of quality reports that gather quality problems [16]. It does not cover quality components offered by DQV, but it can be considered as a complement to DQV for listing errors and inconsistencies found while assessing the quality metrics.

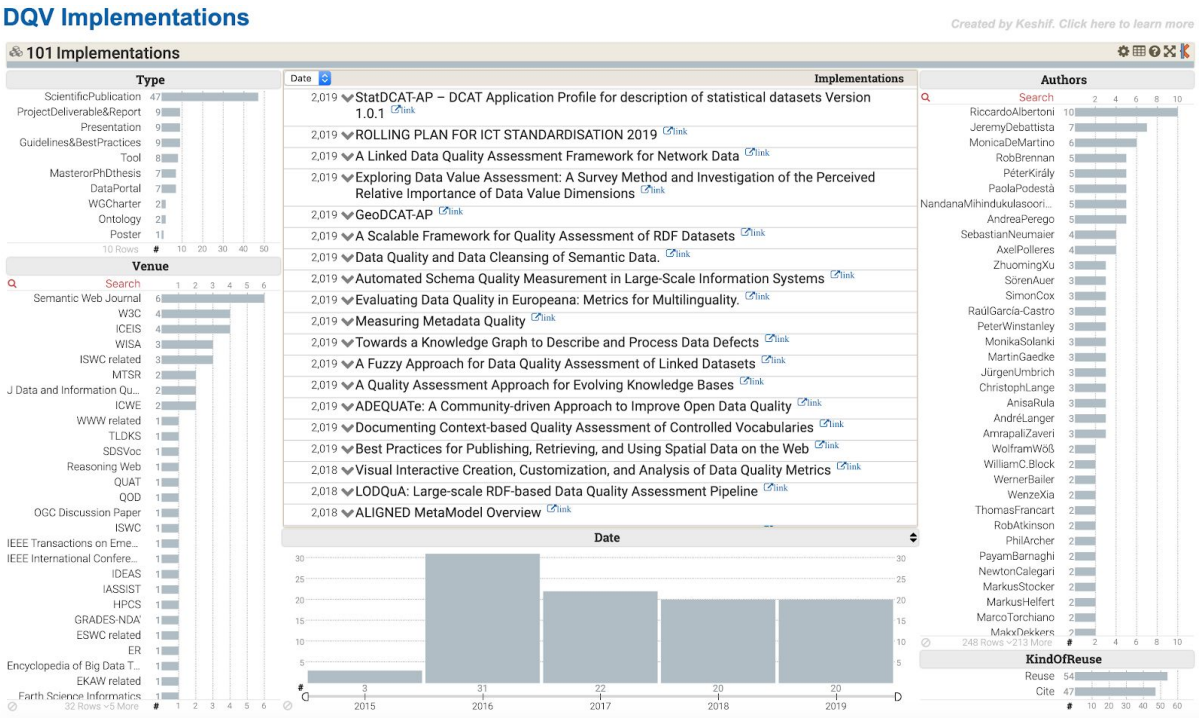

Fig. 2. *DQV implementations collected until September 2019, available at https://w3id.org/quality/DQV/implementations. (Exploration tool derived from the Keshif prototype developed by Mehmet Adil Yalcin [40]; note that dates are not represented in a perfect way, e.g., "2,019" should of course be "2019"*

The Evaluation and Report Language (EARL) [1] is a W3C vocabulary to describe tests and their results in a general setting, which stems from a community effort. EARL is not a direct competitor to DQV, as it has a minimal overlap with its requirements. EARL can be used in the context of quality assessment, and represent some information that could also be represented using DQV. For example, DCAT 2 [6] uses it to represent conformance tests and their results[14], while the latter can be expressed using DQV. This example shows however the difference of scope between EARL and DQV: as noted in section 3.5, DCAT itself uses a different pattern for expressing conformance when complex descriptions of testing activities is not needed, and this pattern (using the dct:conformsTo property[15]) is exactly the one that DQV has adopted. Some EARL notions, such as the one of test result, seem however to generalize notions expressed in DQV. There could be opportunities to align both vocabularies better, e.g., using subclass or subproperty axioms.

There are other works on expressing quality, such as ISO 25010 (SQuaRE) and the ISO/IEC 15939 standards. However, they are not specifically intended for use in linked data contexts. This would require some specific adaptations, and we have listed already in this section the work that we are aware of in this respect. For this reason, we consider further comparison with them out of scope for this review.

In summary, none of the aforementioned vocabularies contemporarily exhibit the DQV characteristics, namely, (1) being the result of a community effort such as a W3C working group; (2) easing interoperability adopting design principles such as minimal ontological commitment and the reuse of established W3C vocabularies; (3) covering a wide spectrum of quality requirements including the representation

[14] https://www.w3.org/TR/vocab-dcat-2/#quality-conformance-test-results

[15] https://www.w3.org/TR/vocab-dcat-2/#Property:resource_conforms_to

of metrics, quality measurements, certificates, and quality annotations.

5. DQV Uptake

The DQV Working Group Note editors, who are the authors of this paper, maintain a list of projects, papers, guidelines and data services reusing DQV[16]. We have gathered the entries in the list by searching for mentions of the DQV namespace and the citations to the DQV Working Group Note [2] and earlier Working Drafts in Google and Google Scholar. DQV implementations collected until September 2019 can be inspected through the exploration tool shown in Fig 2. The categories in which we have organized the implementations can overlap. This is needed to show a multi-faced view of the DQV impact. I.e., we count multiple contributions when a research paper, besides its "theoretical content", also makes available a new ontology or a tool deploying DQV. However, we do not count as a separate items the presentations that are related to a publication already considered.

Table 1: Summary of the DQV implementations collected until September 2019

	Citations and Future Work	Actual Reuses	Total
Data service	0	7	7
Guidelines best practices	3	6	9
Master or PhD thesis	5	2	7
Ontology	0	2	2
Poster	0	1	1
Presentation	4	5	9
Project deliverable/Report	8	1	9
Scientific Publication	24	23	47
Tool	1	7	8
WG charter (not DWBP)	2	0	2
Grand Total	47	54	101

Despite being quite recent, DQV has been referred and reused in more than one hundred entries so far, which are classified in Table 1.

For example, 47 scientific papers have mentioned the DQV (e.g., [3,8,9,15,32,37]), 23 of which have directly reused it (e.g., [4,28,34]). In particular, Radulovic et al. [29] adopt the DQV to model the quality of linked data datasets at different levels of granularity (IRI, statement, graph, dataset). The Aligned project combines DQV with the W3C SHACL Reporting Vocabulary[17], the Test-driven RDF Validation Ontology (RUT)[18] and the Reasoning Violation Ontology (RVO)[19] in order to provide unified quality reports for combined software and data engineering at web-scale [34].

9 international guidelines/best practices suggest DQV for documenting the quality of open data. For example, the StatDCAT Application Profile [17] recommends dqv:QualityAnnotation to document ratings, quality certificates, feedback that can be associated to datasets or distributions. The W3C Spatial Data on the Web Best Practices [36,38] reuses DQV to describe the positional accuracy of spatial data.

8 tools use DQV to encode the results of their elaborations. For instance, qSKOS [25] maps its quality metrics to DQV[20]; RDFUnit provides an API to generate DQV metrics starting from its report[21]; LD sniffer [29] uses a Linked Data Quality ontology (LDQ) which blends the DQV, QMO [30] and EVAL [31] vocabularies in order to document its results.

7 data services have published quality metadata adopting DQV: the Linked Thesaurus fRamework for Environment (LusTRE)[22] encodes in DQV the quality assessment for its thesauri [5]; LODQuator[23] [15] monitors 17 quality metrics on datasets included in the LOD cloud serving results in DQV and daQ; the Open Data Portal Watch[24] harvests the metadata of around 260 Web catalogues and publish quality results along 6 dimensions and 19 metrics [27]; ADEQUATe[25] open data service exploits the DQV to represent quality assessments

[16] https://w3id.org/quality/DQV/implementations

[17] https://www.w3.org/TR/shacl/#validation-report
[18] http://rdfunit.aksw.org/ns/core
[19] http://aligned-project.eu/data/rvo_documentation.html
[20] https://github.com/cmader/qSKOS
[21] https://github.com/AKSW/RDFUnit/tree/master/rdfunit-w3c-dqv
[22] http://linkeddata.ge.imati.cnr.it/
[23] https://w3id.org/lodquator
[24] http://data.wu.ac.at/portalwatch/
[25] http://adequate.at/

and metrics; European Data Portal[26] uses DQV introducing a scoring metric and enhancing catalogue reports; Europeana[27] exploits DQV quality annotations to represent the quality of metadata from its cultural heritage data providers.

International working groups such as the W3C Dataset Exchange Group (DXWG), the RDA WDS/RDA Publishing Data Interest Group and the WDS/RDA Certification of Digital Repositories Interest Group[28] mention DQV in their group charter as a model they should re-use or align with. For example, the DXWG re-uses DQV to document the quality in the latest working draft of the DCAT Revision [6].

6. Conclusion and Future work

DQV is a (meta)data model implemented as an RDF vocabulary, whose original motivation is the documentation of the quality of DCAT Datasets and Distributions. DQV is a community effort developed in the W3C DWBP working group, which gives it high visibility and status. In addition, and more than other proposals for expressing quality information, it specifically embraces design principles meant to favor its reusability and uptake. The adoption of minimal ontological commitment has led us to avoid unnecessary domain restrictions, for DQV can be applied to any kind of web resource, not only DCAT Datasets and Distributions. The reuse of consolidated design patterns from other standard vocabularies has minimized the number of new terms defined in DQV and it is expected to shorten its learning curve. These factors seem to have facilitated a number of DQV reuses, which is encouraging considering the recency of DQV.

As DQV is a Working Group Note, it is not a final recommendation and can be seen as a work in progress. As its editors, we are committed to support the adoption of DQV and consider issues and questions arising from the reuse of DQV in specific use cases and projects. We are especially interested in feedback from DQV adopters about barriers or requirements, which might have been disregarded in this first specification round. From the feedback received so far, we are considering the following future activities:

- define a default ShEx schema and/or SHACL shape to help adopters to understand the (few) constraints that apply by default to DQV data and potentially help them to create their own profiles/extensions of DQV, including additional constraints that their applications may need;
- publish a JSON-LD context [22] to facilitate the use of DQV in a JSON environment;
- include a notion of severity for the discovered quality issues;
- define DQV mappings with metadata models (or extensions of such models with DQV elements) adopted in domain specific portals, such as INSPIRE;
- develop consumption tools such as a visualizer;
- develop registries (possibly equipped with APIs) for the dimensions and metrics coming from different quality frameworks and the alignments between them.

Besides these, some already started work is likely to bring new lines of activity around DQV: The ongoing DCAT revision carried out within the W3C Data Exchange Working Group [6] explicitly considers DQV providing examples and guidance to document dataset and distribution quality. In addition, the recently launched Google Dataset search[29] and the related mapping of DCAT with schema.org raises new opportunities for DQV, which could as well be proposed for mapping into Schema.org.

Acknowledgement

The authors thank Jeremy Debattista for contributing to the design of Quality Measurement, and for his support in understanding the daQ Ontology; Nandana Mihindukulasooriya for contributing to Quality Policy; Amrapali Zaveri for mapping the ISO and linked data quality dimensions; Makx Dekkers and Christophe Guéret

[26] https://www.europeandataportal.eu/

[27] https://www.europeana.eu/

[28] https://www.rd-alliance.org/sites/default/files/case_statement/DataPublication-FitnessForUse-CaseStatement.pdf, Accessed 2018-09-01

[29] https://toolbox.google.com/datasetsearch

for driving the discussions in the early stage of DQV specification. They also thank the DWBP Group chairs Hadley Beeman, Yaso Córdova, Deirdre Lee, the staff contact Phil Archer, and gratefully acknowledge the contributions made to the DQV discussion by all members of the working group and external commenters, in particular, the contributions received from Andrea Perego, Ghislain Auguste Atemezing, Carlos Laufer, Annette Greiner, Michel Dumontier, Eric Stephan.

Bibliography

[1] S. Abou-Zahra, "Evaluation and Report Language (EARL) 1.0 Schema". W3C. 2017. W3C Working Group Note. https://www.w3.org/TR/EARL10-Schema/

[2] R. Albertoni and A. Isaac, "Data on the Web Best Practices: Data Quality Vocabulary", W3C Working Group Note. 2016, https://www.w3.org/TR/vocab-dqv/.

[3] R. Albertoni, M. De Martino, and P. Podestà, "Quality measures for skos:ExactMatch linksets: an application to the thesaurus framework LusTRE", Data Techn. Applic., vol. 52, no. 3, pp. 405–423, 2018.

[4] R. Albertoni, M. De Martino, and A. Quarati, "Documenting Context-based Quality Assessment of Controlled Vocabularies", IEEE Trans. Emerg. Top. Comput., to appear.

[5] R. Albertoni, M. De Martino, P. Podestà, A. Abecker, R. Wössner, and K. Schnitter, "LusTRE: a framework of linked environmental thesauri for metadata management", Earth Sci. Informatics, vol. 11, no. 4, pp. 525-544, 2018.

[6] R. Albertoni, D. Browning, S. Cox, A. Gonzalez Beltran, A. Perego, P. Winstanley "Data Catalog Vocabulary (DCAT) - Version 2", W3C Candidate Recommendation. 03 October 2019 https://www.w3.org/TR/vocab-dcat-2/

[7] T. Baker, S. Bechhofer, A. Isaac, A. Miles, G. Schreiber, and E. Summers, "Key choices in the design of Simple Knowledge Organization System (SKOS)", Web Semant. Sci. Serv. Agents World Wide Web, vol. 20, pp. 35–49, 2013.

[8] W. Beek, F. Ilievski, J. Debattista, S. Schlobach, and J. Wielemaker, "Literally better: Analyzing and improving the quality of literals", Semant. Web, vol. 9, no. 1, pp. 131–150, 2017.

[9] M. Ben Ellefi et al., "RDF dataset profiling – a survey of features, methods, vocabularies and applications", Semant. Web, vol. 9, no. 5, pp. 677–705, 2018.

[10] N. Calegari, B. F. Loscio, and C. Burle, "Data on the Web Best Practices", W3C Recommendation, 2017, https://www.w3.org/TR/dwbp/.

[11] P. Ciccarese, B. Young, and R. Sanderson, "Web Annotation Vocabulary", W3C Recommendation. 2017, https://www.w3.org/TR/annotation-vocab/.

[12] R. Cyganiak and D. Reynolds, "The RDF Data Cube Vocabulary", W3C Recommendation. 2014 http://www.w3.org/TR/vocab-data-cube/.

[13] J. Debattista, C. Lange, and S. Auer, "Representing dataset quality metadata using multi-dimensional views", in Proceedings of the 10th International Conference on Semantic Systems - SEM '14, 2014, pp. 92–99.

[14] J. Debattista, C. Lange, and S. Auer, "daQ, an Ontology for Dataset Quality Information", in Proceedings of the Workshop on Linked Data on the Web co-located with the 23rd International World Wide Web Conference (WWW 2014), Seoul, Korea, April 8, 2014, vol. 1184.

[15] J. Debattista, C. Lange, S. Auer, and D. Cortis, "Evaluating the quality of the LOD cloud: An empirical investigation", Semant. Web, vol. 9, no. 6, pp. 859–901, Sep. 2018.

[16] J. Debattista, C. Lange, and S. Auer, "Luzzu - A Framework for Linked Data Quality Assessment", Tenth IEEE International Conference on Semantic Computing (ICSC), Laguna Hills, CA, USA, February 4-6, 2016

[17] M. Dekkers, C. Nelson, S. Kotoglou, and F. Barthelemy, "StatDCAT-AP – DCAT Application Profile for description of statistical datasets", 2016, https://joinup.ec.europa.eu/node/157143.

[18] European Commission, "GeoDCAT-AP : A

geospatial extension for the DCAT application profile for data portals in Europe Version 1.0", pp. 1–59, 2015 https://joinup.ec.europa.eu/node/148281.

[19] C. Fürber and M. Hepp, "Towards a vocabulary for data quality management in semantic web architectures", Proc. 1st Int. Work. Linked Web Data Manag. - LWDM '11, p. 1, 2011.

[20] ISO/IEC, "ISO/IEC 25012 (2008) - Software product Quality Requirements and Evaluation (SQuaRE) - Data quality model", International Standard ISOIEC 25012, 2008. [Online]. Available: http://iso25000.com/index.php/en/iso-25000-standards/iso-25012.

[21] J. E. Labra Gayo, E. Prud'hommeaux, I. Boneva, D. Kontokostas, "Validating RDF Data", Synthesis Lectures on the Semantic Web: Theory and Technology, Vol. 7, No. 1, 1-328, Morgan & Claypool, 2018. http://book.validatingrdf.com/

[22] M. Lanthaler, G. Kellogg, M. Sporny, "JSON-LD 1.0", W3C Working Draft, 2014, http://www.w3.org/TR/json-ld/.

[23] B. F. Loscio, D. Lee, and P. Archer, "Data on the Web Best Practices Use Cases & Requirements", W3C Working Group Note. 2015 http://www.w3.org/TR/dwbp-ucr/.

[24] F. Maali and J. Erickson, "Data Catalog Vocabulary (DCAT)", W3C Recommendation. 2014 http://www.w3.org/TR/vocab-dcat/.

[25] C. Mader, B. Haslhofer, and A. Isaac, "Finding quality issues in SKOS vocabularies", Theory Pract. Digit. Libr., 2012.

[26] A. Miles and S. Bechhofer, "SKOS Simple Knowledge Organization System Reference", W3C Recommendation, August., 2009 http://www.w3.org/TR/skos-reference/.

[27] S. Neumaier, A. Polleres, S. Steyskal, and J. Umbrich, "Data Integration for Open Data on the Web", in Reasoning Web. Semantic Interoperability on the Web: 13th International Summer School 2017, London, UK, July 7-11, 2017, Tutorial Lectures, G. Ianni, D. Lembo, L. Bertossi, W. Faber, B. Glimm, G. Gottlob, and S. Staab, Eds. Cham: Springer International Publishing, 2017, pp. 1–28.

[28] S. Neumaier, J. Umbrich, and A. Polleres, "Lifting Data Portals to the Web of Data", in Workshop on Linked Data on the Web co-located with 26th International World Wide Web Conference (WWW 2017), 2017.

[29] F. Radulovic, N. Mihindukulasooriya, R. García-Castro, and A. Gómez Pérez, "A comprehensive quality model for Linked Data", Semant. Web, vol. 9, no. 1, pp. 3–24, 2018.

[30] F. Radulovic, R. García-Castro, "The Quality Model Ontology", 05 August 2015, http://purl.org/net/QualityModel#.

[31] F. Radulovic, R. García-Castro, "The Evaluation Result Ontology", 05 August 2015, http://purl.org/net/EvaluationResult#.

[32] M. Rashid, M. Torchiano, G. Rizzo, and N. Mihindukulasooriya, "A Quality Assessment Approach for Evolving Knowledge Bases" Semant. Web., vol 10, no. 2, pp. 349-383, 2019.

[33] S. Sahoo, D. McGuinness, and T. Lebo, "PROV-O: The PROV Ontology", W3C Recommendation, April 2013 http://www.w3.org/TR/prov-o/.

[34] M. Solanki, B. Božić, M. Freudenberg, D. Kontokostas, C. Dirschl, and R. Brennan, "Enabling Combined Software and Data Engineering at Web-Scale: The ALIGNED Suite of Ontologies BT - The Semantic Web – ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part II", P. Groth, E. Simperl, A. Gray, M. Sabou, M. Krötzsch, F. Lecue, F. Flöck, and Y. Gil, Eds. Cham: Springer International Publishing, 2016, pp. 195–203.

[35] E. Stephan, S. Purohit, and B. F. Loscio, "Data on the Web Best Practices: Dataset Usage Vocabulary", W3C Working Group Note. 2016, https://www.w3.org/TR/vocab-duv/.

[36] J. Tandy, P. Barnaghi, and L. van den Brink, "Spatial Data on the Web Best Practices", W3C Working Group Note, 2017 https://www.w3.org/TR/sdw-bp/.

[37] V. Theodorou, I. Gerostathopoulos, S. Amini,

R. Scandariato, C. Prehofer, and M. Staron, “Theta Architecture: Preserving the Quality of Analytics in Data-Driven Systems”, in New Trends in Databases and Information Systems, 2017, pp. 186–198.

[38] L. van den Brink et al., “Best practices for publishing, retrieving, and using spatial data on the web”, Semant. Web,vol. 10, no. 1, pp. 95-114, 2019.

[39] S. Villata and R. Iannella, “ODRL Information Model”, W3C Working Draft. 2017 https://www.w3.org/TR/odrl-model/.

[40] Yalçin, M. A., Elmqvist, N., and Bederson, B. B., “Keshif: Rapid and Expressive Tabular Data Exploration for Novices”, IEEE Trans. Vis. Comput. Graph., vol. 24, no. 8, pp. 2339–2352, 2018.

[41] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann, and S. Auer, “Quality Assessment for Linked Data: A Survey”, Semant. Web J., vol. 1, no. 7, pp. 63–93, 2016.