

## 1. Quality Characteristics and Temporal Analysis

Data quality is a cross-disciplinary and multidimensional concept. Data quality, in general, relates to the perception of the "fitness for use" in a given context [8]. According to Pipino *et al.* [6], based on context, quality can be both subjective perceptions and objective measurements. This means that data quality is dependent on the actual use case.

In our approach, we use two data quality standard reference frameworks: ISO/IEC 25012 [4] and W3C DQV [3]. ISO/IEC 25012 [4] defines a general data quality model for data retained in a structured format within a computer system. This model defines the quality of a data product as the degree to which data satisfies the requirements set by the product owner organization. The W3C Data on the Web Best Practices Working Group has been chartered to create a vocabulary for expressing data quality<sup>1</sup>. The Data Quality Vocabulary (DQV) is an extension of the DCAT vocabulary<sup>1</sup>. It covers the quality of the data, how frequently is it updated, whether it accepts user corrections, and persistence commitments.

Besides, to further compare our selected quality characteristics<sup>2</sup> we explored the foundational work on the linked data quality by Zaveri *et al.* [11]. They surveyed existing literature and identified a total of 18 different data quality dimensions (criteria) applicable to linked data quality assessment.

Since the measurement terminology suggested in the two standards is different, we briefly summarize the one adopted in this paper and the relative mapping as reported in Table 1.

### 1.1. Quality Issues

A data quality issues is a set of anomalies that can affect the potentiality of the applications that use the data [1]. We can identify quality issues through data quality measure. A data quality measure variable to which a value is assigned as the result of measurement of data quality characteristic [4]. Assessing the quality of data usually requires a large number of quality measures to be computed [2]. Each of the quality characteristics identifies a specific set of quality issues. In this paper, we focused on three main quality issues of

Table 1  
Measurement terminology

Definition	ISO 25012	W3C DQV
Category of quality attributes	Characteristic	Dimension
Variable to which a value is assigned as the result of a measurement function applied to two or more measure elements	Measure	Metric
Variable defined in terms of an attribute and the measurement method for quantifying it	Measure Element	-
Numerical value that characterize a quality feature	Value	Observation
Set of operations having the object of determining a value of a measure	Measurement	Measurement

a knowledge baase such as (i) Consistency, (ii) Completeness, and (iii) Persistency.

**Consistency** relates to a fact being inconsistent in a KB. In particular, inconsistency relates to the presence of unexpected properties.

As an example let us consider a DBpedia resource of type *foaf:Person: X. Henry Goodnough*<sup>3</sup>. We find as expected a *dbo:birthDate* property, but we unexpectedly find the property *dbo:Infrastructure/length*. This is a clear inconsistency: in fact according to the ontology we can expect the latter property for a resource of type *dbo:Infrastructure*, not for a person.

To better understand where the problem lies, we need to look at the corresponding Wikipedia page<sup>4</sup>. Even though the page reports the information about an engineer who graduated from Harvard, it contains an info-box, shown in Figure 1, that refers to a dam, the Goodnough Dike. The inconsistency issue derives from the data present in the source page that resulted into the resource being typed both as a person and as a piece of infrastructure. We can expect such kind of structure to be fairly rare – in fact the case we described is the only case of a person with a *dbo:Infrastructure/length* property – and can be potentially detected by looking at the frequency of the predicates within a type of resource. For instance for the resources of type *foaf:Person* there are 1035 distinct

<sup>1</sup><https://www.w3.org/TR/prov-o>

<sup>2</sup>In our work we will identify the quality aspects using the term quality characteristics from ISO-25012 [4] that corresponds to the term quality dimension from DQV [3].

<sup>3</sup>[http://dbpedia.org/resource/X.\\_Henry\\_Goodnough](http://dbpedia.org/resource/X._Henry_Goodnough)

<sup>4</sup>[https://en.wikipedia.org/wiki/X.\\_Henry\\_Goodnough](https://en.wikipedia.org/wiki/X._Henry_Goodnough)

## X. Henry Goodnough

From Wikipedia, the free encyclopedia

**X. Henry Goodnough**, (1860–1935), engine

Goodnough Dike	
	
Goodnough Dike the wet side	
Official name	Goodnough Dike
Location	Ware
Coordinates	<span><span><span><span><span>42°17′51″N</span> <span>72°17′56″W</span></span></span><span><span>﻿</span> / <span>﻿</span></span><span><span>42.29750°N 72.29889°W</span><span><span>﻿</span> / <span>42.29750; -72.29889</span></span></span></span></span>
Construction began	1933
Opening date	1938
Operator(s)	MWRA
Dam and spillways	
Impounds	Beaver Brook
Height	264 <span> </span> ft (80.47 <span> </span> m)
Length	2,140 <span> </span> ft (652.3 <span> </span> m)
Width (base)	878 <span> </span> ft (267.61 <span> </span> m)
Reservoir	
Creates	Quabbin Reservoir

Fig. 1. Example of inconsistent Wikipedia data.

predicates, among which 142 occur only once for DBpedia version 201604.

**Completeness** relates to the resources or properties missing from a knowledge base. This happens when information is missing or has been removed.

As an example, let us consider a DBpedia resource of type *dbo:Person/Astronauts*: *Abdul Ahad Mohmand*<sup>5</sup>. When looking at the source Wikipedia page<sup>6</sup>, we observe that, as shown in Figure 2, the infobox reports a “Time in space” datum. The DBpedia ontology includes a *dbo:Astronaut/TimeInSpace* and several other astronauts have that property, but the resource we consider is missing it.

While it is generally difficult to spot that kind of incompleteness, for the case under consideration it is easier because that property was present for the resource under consideration in the previous version of DBpedia, i.e. the 2015-10 release. It is an incompleteness introduced by the evolution of the knowledge base. It can be spotted by looking at the frequency of predicates inside a resource type. In particular, in the release of 2016-04 there are 419 occurrences of the *dbo:Astronaut/TimeInSpace* predicate over 634 astro-

<sup>5</sup>[http://dbpedia.org/resource/Abdul\\_Ahad\\_Mohmand](http://dbpedia.org/resource/Abdul_Ahad_Mohmand)

<sup>6</sup>[https://en.wikipedia.org/wiki/Abdul\\_Ahad\\_Mohmand](https://en.wikipedia.org/wiki/Abdul_Ahad_Mohmand)


Abdul Ahad Mohmand	
Intercosmos Research Cosmonaut	
Nationality	Afghan
Status	Retired
Born	January 1, 1959 (age 58) Sardah, Afghanistan
Other occupation	Pilot
Alma mater	Kabul University
Rank	Colonel
Time in space	8d 20h 26min
Selection	1988
Missions	Mir EP-3 (Soyuz TM-6/Soyuz TM-5)
Mission insignia	

Fig. 2. Example of incomplete Wikipedia data.

naut resources, while in the previous version they were 465 out of 650 astronauts.

**Persistency** relates to resources that were present in a previous KB release but they disappeared. This happens when information has been removed. As an example let us consider a 3cixty Nice resource of type *lode:Event* that has as label the following: “Modéliser, piloter et valoriser les actifs des collectivités et d’un territoire grâce aux maquettes numériques: retours d’expériences et bonnes pratiques”<sup>7</sup>. This resource happened to be part of the 3cixty Nice KB since it has been created the first time, but in a subsequent release it got removed even though it should not have been removed.

Such a problem is generally complex to be traced manually because it requires a per-resource check over the different releases. It can, instead, be spotted by looking at the total frequency of entities of a given resource type. In particular in the investigated example taken from the 3cixty Nice KB released on 2016-09-09, we have observed an unexpected drop of resources of the type event from the previous release dated as 2016-06-15, which has triggered further investigations.

### 1.2. Temporal Analysis

Knowledge bases are often maintained by large communities that act as curators to ensure their quality [10]. Knowledge base changes can be categorized as follows: *i*) resource representations and links that are created, updated and removed; *ii*) the entire graph can change or disappear [7]. The kind of evolution that

<sup>7</sup><http://data.linkedevents.org/event/006dc982-15ed-47c3-bf6a-a141095a5850>

```

Subject Item
  n2:006dc982-15ed-47c3-bf6a-a141095a5850
rdf:type
  lode:Event
rdfs:label
  Modéliser, piloter et valoriser les actifs des collectivités et d'un territoire grâce aux
  maquettes numériques : retours d'expériences et bonnes pratiques
rdfs:seeAlso
  n13:en
cixty:descriptionScore
  0.0
cixty:posterScore
  1.0
lode:poster
  n4:006dc982-15ed-47c3-bf6a-a141095a5850
dc:identifier
  MN13
dc:publisher
  n14:com
locationOnt:businessType
  n15:event
lode:atPlace
  n12:be7fac75-bb59-41fd-a626-4bd7e77f0a7f
lode:atTime
  n6:interval
lode:hasCategory
  Conférences Maquette Numérique
lode:inSpace
  n6:geometry
lode:involvedAgent
  n11:a40c9900f85a517cef40ef8f1e4289b9 n11:7f1a9cc96861920e147505e23ea4f913
  n11:dce31cbcfdad5c0a180fb4d0efd0c511
locationOnt:cell
  n9:1301

```

Fig. 3. Example of a 3cixty Nice KB resource that unexpectedly disappeared from the release of 2016-06-15 to the other 2016-09-09.

a KB is subjected to depends on several factors, such as:

- Frequency of update: KBs can be updated almost continuously (e.g. daily or weekly) or at long intervals (e.g. yearly);
- Application area: depending on the specific domain, updates can be minor or substantial. For instance, social data is likely to change more frequently than encyclopedic data;
- Data acquisition: the process used to acquire the data to be stored in the KB and the characteristics of the sources may influence the evolution; For instance, updates on individual resources cause minor changes when compared to a complete reorganization of a data source's infrastructure such as a change of the domain name;
- Link between data sources: when multiple sources are used for generating a KB, the alignment and compatibility of such sources affect the overall KB evolution. The differences of KBs have been proved to play a crucial role in various curation tasks such as the synchronization of autonomously developed KB versions, or the visualization of the evolution history of a KB [5] for more user-friendly change management.

### 1.3. Temporal-based Quality characteristics and Measures

In this section, we define four temporal quality characteristics that allow addressing the aforementioned issues.

Zaveri et al. [11] classified quality dimensions into four groups: *i*) intrinsic, those that are independent of the users context; *ii*) contextual, those that highly depend on the context of the task at hand, *iii*) representational, those that capture aspects related to the design of the data, and *iv*) accessibility, those that involve aspects related to the access, authenticity and retrieval of data obtain either the entire or some portion of the data (or from another source) for a particular use case. The quality dimensions we propose fall into the groups of intrinsic and representational. Our approach focuses on two different types of elements in a KB: subjects and predicates. The objects, either resources or literals, are not considered. Concerning the subjects we consider them collectively by grouping according to the class – or a property defined as *rdf:type* – they belong to. As far as properties are concerned, we analyze separately the properties of the resources of a given class.

Table 2 reports the proposed characteristics, along with the quality issue they address, the level of measure – either class subject or property –, and the corresponding quality characteristic as defined in the ISO 25012 standard.

Table 2  
Quality Characteristics in KB evolution.

Quality Issues	ISO/IEC 25012	Levels	Quality Characteristics
Persistency	Credibility	Class	Persistency
Persistency	Efficiency	Class	Historical Persistency
Completeness	Completeness	Property	Completeness
Consistency	Consistency	Class & Property	Consistency

In order to measure the degree to what extent a certain data quality characteristics is fulfilled for a given KB, each characteristics is formalized and expressed in terms of a measure with a value in the range [0, 1]. We call this measurement function for a data quality characteristics.

### 1.3.1. Basic Measure Elements

In our approach, we consider changes at the statistical level in terms of variation of absolute and relative frequency count of subjects and predicates between pairs of KB versions.

Our approach shares the same basis as Papavasiliou *et al.* [5]. They divided the changes into *Low-Level* and *High-Level*. In our evaluation approach, we focus on *Low-Level* changes that consist in the addition or deletion of a triple from a KB.

In particular we aim to detect changes in two basic statistical measures that can be computed with the most simple operation, i.e. counting. The computation is performed on the basis of the classes in a KB ( $V$ ), i.e. given a class  $C$  we consider all the triples  $t$  whose subjects have the type  $C$ .

The first measure element we define is the count of the instances of a class  $C$ :

$$\text{count}(C) = |\{s : \exists \langle s, \text{typeof}, C \rangle \in V\}|$$

The  $\text{count}(C)$  measurement can be performed by means of a basic SPARQL query such as:

```
SELECT COUNT(DISTINCT ?s) AS ?COUNT
WHERE { ?s a <C> . }
```

The second measure element focuses on the frequency of the predicates, within a class  $C$ . We define the frequency of a predicate (in the scope of class  $C$ ) as:

$$\text{freq}(p, C) = |\{(s, p, o) \in V : \exists \langle s, \text{typeof}, C \rangle \in V\}|$$

The  $\text{freq}(p, C)$  measurement can be performed by means of a simple SPARQL query having the following structure:

```
SELECT COUNT(*) AS ?FREQ
WHERE {
  ?s <p> ?o.
  ?s a <C>.
}
```

There is an additional basic measure element that can be used to build derived measures: the number of predicates present for the subject class  $C$  in the release  $i$  of the KB.

$$NP(C) = |\{p : \exists \langle s, p, o \rangle \in V \wedge \langle s, \text{typeof}, C \rangle \in V\}|$$

The  $NP(C)$  measure can be collected by means of a SPARQL query having the following structure:

```
SELECT COUNT(DISTINCT ?p) AS ?NP
WHERE {
  ?s ?p ?o.
  ?s a <C>.
}
```

The essence of the proposed approach is the comparison of distinct releases of a KB with respect to subject count or predicate frequency measures. We will use a subscript to indicate the release the measure refers to. The releases are numbered progressively as integers and, by convention, the most recent release is  $n$ . So, for instance,  $\text{count}_{n-1}(\text{foaf:Person})$  represents the count of subjects typed with *foaf:Person* in the last release of the knowledge base under consideration.

### 1.3.2. Persistency

This quality characteristic relates to the Credibility quality characteristic in the ISO/IEC 25012 standard. Credibility is the “degree to which data has attributes that are regarded as true and believable by users in a specific context of use. Credibility includes the concept of authenticity (the truthfulness of origins, attributions, commitments)” [4]. Considering the correspondence presented in W3C DQV, Zaveri *et al.* [11] implies Credibility as Trustworthiness. They report that “Trustworthiness is defined as the degree to which the information is accepted to be correct, true, real and credible.”

An additional important feature to be considered when analyzing knowledge base is that the information stored is expected to grow, either because of new facts appearing in the reality, as time passes by, or due to an extended scope coverage [9]. Persistency measures provides an indication of the adherence of a knowledge base to such continuous growth assumption. Using this quality measure, data curators can identify the classes for which the assumption is not verified.

The *Persistency* of a class  $C$  in a release  $i : i > 1$  is defined as:

$$\text{Persistency}_i(C) = \begin{cases} 1 & \text{if } \text{count}_i(C) \geq \text{count}_{i-1}(C) \\ 0 & \text{if } \text{count}_i(C) < \text{count}_{i-1}(C) \end{cases}$$

the value is 1 if the count of subjects of type  $C$  is not decreasing, otherwise it is 0.

Persistency at the knowledge base level, i.e. when all classes are considered, can be computed as the proportion of persistent classes:

$$Persistence_i = \frac{\sum_{j=1}^{NC} Persistence_i(C_j)}{NC}$$

where  $NC$  is the number of classes analyzed in the KB.

### 1.3.3. Historical Persistency

This quality characteristic relates to the Efficiency quality characteristic defined in the ISO/IEC 25012 standard. Efficiency is defined as the "degree to which data has attributes that can be processed and provide the expected levels of performance by using the appropriate amounts and types of resources in a specific context of use" [4].

Considering the mapping presented in W3C DQV, Zaveri et al. [11] implies Efficiency as Performance. They define as, "Performance refers to the efficiency of a system that binds to a large dataset, that is, the more performing a data source is the more efficiently a system can process data".

Historical persistency is a derived measurement function using the persistency measure over all releases of KB. Historical persistency dimensions explore entire KB evolution for a specific entity to detect inconsistency. This metric extends the persistency metric to provide insights on the series of KB releases. It considers all entities presented in a KB and give an overview of the KB. Data curators can get an overview of knowledge base persistency issues over all releases. It helps data curators to decide which knowledge base release can be used for future data management tasks.

The Historical Persistency measure evaluates the persistency over the history of the KB and is computed as the average of the pairwise persistency measures for all releases.

$$H\_Persistence(C) = \frac{\sum_{i=2}^n Persistence_i(C)}{n - 1}$$

Similarly to Persistency, it is possible to compute Historical Persistency at the KB level:

$$H\_Persistence = \frac{\sum_{i=2}^n Persistence_i}{n - 1}$$

### 1.3.4. Consistency

This quality characteristic relates to ISO/IEC 25012 standard Consistency quality characteristic.

The Consistency is defined as the "degree to which data has attributes that are free from contradiction and are coherent with other data in a specific context of use. It can be either or both among data regarding one entity and across similar data for comparable entities" [4].

Considering the correspondence presented in W3C DQV, Zaveri et al. mention Consistency as Conciseness. They define as, "Conciseness refers to the minimization of redundancy of entities at the schema and the data level."

We assume that extremely rare predicates are potentially inconsistent, see e.g. the *dbo:Infrastructure/length* property discussed in the example presented in Section 1.1. We can evaluate the consistency of a predicate on the basis of the frequency basic measure.

We define the consistency of a property  $p$  in the scope of a class  $C$ :

$$Consistency_i(p, C) = \begin{cases} 1 & \text{if } \text{freq}_i(p, C) \geq T \\ 0 & \text{if } \text{freq}_i(p, C) < T \end{cases}$$

Where  $T$  is a threshold that can be either a KB-dependent constant<sup>8</sup> or can be defined on the basis of the count of the scope class, e.g.  $T = \text{count}(C)/10000$ .

### 1.3.5. Completeness

The ISO/IEC 25012 defines the Completeness quality characteristic as the "degree to which subject data associated with an entity has values for all expected attributes and related entity instances in a specific context of use" [4].

In Zaveri et al., Completeness refers to the degree to which all required information is present in a particular dataset. In terms of Linked Data, completeness comprises of the following aspects: *i*) Schema completeness, the degree to which the classes and properties of an ontology are represented, thus can be called "ontology completeness"; *ii*) Property completeness, measure of the missing values for a specific property, *iii*) Population completeness is the percentage of all real-world objects of a particular type that are represented in the datasets, and *iv*) Interlinking completeness, which has to be considered especially in Linked

<sup>8</sup>In our experiments we used  $T=100$  as empirically verified to maximize the precision of the approach in detecting quality issues.

Data, refers to the degree to which instances in the dataset are interlinked.

Temporal-based completeness focuses on the removal of information as a negative effect of the KB evolution. It is based on the continuous growth assumption as well; as a consequence we expect properties of subjects should not be removed as the KB evolves (e.g. *dbo:Astronaut/TimeInSpace* property described in the example presented in Section 1.1).

The basic measure we use is the frequency of predicates, in particular, since the variation in the number of subjects can affect the frequency, we introduce a normalized frequency as:

$$Nf_i(p, C) = \frac{\text{freq}_i(p, C)}{\text{count}_i(C)}$$

On the basis of this derived measure we can thus define completeness of a predicate  $p$  in the scope of a class  $C$  as:

$$Completeness_i(p, C) = \begin{cases} 1, & Nf_i(p, C) \geq Nf_{i-1}(p, C) \\ 0, & Nf_i(p, C) < Nf_{i-1}(p, C) \end{cases}$$

At the class level the completeness is the proportion of complete predicates and can be computed as:

$$Completeness_i(C) = \frac{\sum_{k=1}^{NP_i(C)} Completeness_i(p_k, C)}{NP_i(C)}$$

where  $NP_i(C)$  is the number of predicates present for the subject class  $C$  in the release  $i$  of the knowledge base, and  $p_k$ .

## References

- [1] Sören Auer, Jens Lehmann, and Axel-Cyrille Ngonga Ngomo. Introduction to linked data and its lifecycle on the web. In *Proceedings of the 7th International Conference on Reasoning Web: Semantic Technologies for the Web of Data*, RW'11, pages 1–75, Berlin, Heidelberg, 2011. Springer-Verlag.
- [2] Jeremy Debattista, Sören Auer, and Christoph Lange. Luzzu methodology and framework for linked data quality assessment. *Journal of Data and Information Quality (JDIQ)*, 8(1):4, 2016.
- [3] Antoine Isaac and Riccardo Albertoni. Data on the web best practices: Data quality vocabulary. W3C note, W3C, December 2016. <https://www.w3.org/TR/2016/NOTE-vocab-dqv-20161215/>.
- [4] ISO/IEC. 25012:2008 – software engineering – software product quality requirements and evaluation (square) – data quality model. Technical report, ISO/IEC, 2008.
- [5] Vicky Papavasileiou, Giorgos Flouris, Irini Fundulaki, Dimitris Kotzinos, and Vassilis Christophides. High-level change detection in rdf (s) kbs. *ACM Transactions on Database Systems (TODS)*, 38(1):1, 2013.
- [6] Leo L. Pipino, Yang W. Lee, and Richard Y. Wang. Data quality assessment. *Commun. ACM*, 45(4):211–218, April 2002.
- [7] Yannis Roussakis, Ioannis Chrysakis, Kostas Stefanidis, Giorgos Flouris, and Yannis Stavarakas. A flexible framework for understanding the dynamics of evolving rdf datasets. In *International Semantic Web Conference*, pages 495–512. Springer, 2015.
- [8] Giri Kumar Tayi and Donald P Ballou. Examining data quality. *Communications of the ACM*, 41(2):54–57, 1998.
- [9] Jürgen Umbrich, Stefan Decker, Michael Hausenblas, Axel Polleres, and Aidan Hogan. Towards dataset dynamics: Change frequency of linked open data sources. 2010.
- [10] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.
- [11] Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, Jens Lehmann, and Sören Auer. Quality Assessment for linked Data: A Survey. *Semantic Web*, 7(1):63–93, 2016.