Undefined 0 (2015) 1–0
IOS Press

Quality Assurance of RDB2RDF Mappings

Patrick Westphal [a] Claus Stadler [a] Jens Lehmann [a]
[a] *AKSW, Department of Computer Science, University of Leipzig, Germany,*
E-mail: {pwestphal | cstadler | lehmann}@informatik.uni-leipzig.de

Abstract. Since datasets in the Web of Data stem from many different sources, ranging from automatic extraction processes to extensively curated knowledge bases, their quality also varies. Thus, significant research efforts were made to measure and improve the quality of Linked Open Data. Nevertheless, those approaches suffer from two shortcomings: First, most quality metrics are insufficiently formalised to allow an unambiguous implementation which is required to base decision on them. Second, they do not take the creation process of RDF data into account. A popular extraction approach is the mapping of relational databases to RDF (RDB2RDF). RDB2RDF techniques allow to create large amounts of RDF data with only few mapping definitions. This also means that single errors in an RDB2RDF mapping can affect a considerable portion of the generated data. In this paper we present an approach to assess RDB2RDF mappings also considering the actual process of the RDB to RDF transformation. This allows to detect and fix problems at an earlier stage before resulting in potentially thousands of data quality issues in published data. We propose a formal model and methodology for the evaluation of the RDB2RDF mapping quality and introduce actual metrics. We evaluate our assessment framework by applying our reference implementation on different real world RDB2RDF mappings.

1. Introduction

The Web of Data comprises datasets stemming from many different sources, such as crowd-sourced user input, automatic extraction processes or extensively curated knowledge bases. Accordingly, the quality of the corresponding datasets also differs: Whereas crowd-sourced approaches and automatic extraction processes are usually prone to errors, data compiled by domain experts and knowledge engineers can be expected to have high quality on average. In order to measure the status quo of Semantic Web data with regards to its quality, different research efforts have been made (see [34] for a survey). Besides conceptual and theoretical considerations, several tools and methodologies for practical assessments were proposed [7,22,17,16]. Despite this body of research work, quality assessment lacks solid formal foundations. In many cases, quality assessment metrics have been proposed in an ad hoc manner[1] thereby leaving room for ambiguous interpretation and implementation as well as constituting an obstacle for semantically equivalent but more efficient implementations of metrics. In this work, we aim to build this foundation by formally defining 43 metrics of which only a subset is presented in this article for space reasons. All definitions and details are available and described in the accompanying technical report[2].

At the moment, many knowledge bases in the Web of Data are not manually created, but automatically derived via mass data generation approaches. Even though knowledge bases maintained by domain experts are preferable to automatically extracted data as far as their quality is concerned, such mass generation approaches are valuable contributions to the Web of Data due to the amount of potentially useful data they generate. One of the most prominent mass extraction approaches is the the mapping of relational databases to RDF (RDB2RDF). With RDB2RDF techniques large amounts of RDF data can be generated

[1] An exception are metrics related to logical reasoning, e.g. detection of inconsistencies and incoherences, which are based on several decades of research in reasoning in description logics.

[2] `http://svn.aksw.org/papers/2014/report_QA_RDB2RDF/public.pdf`

0000-0000/15/$00.00 © 2015 – IOS Press and the authors. All rights reserved

with only few mapping definitions. Nonetheless, this also means that single errors in an RDB2RDF mapping can affect a considerable portion of the created data. Unfortunately, such errors are not easily detectable by general purpose quality assessment approaches that only consider the generated output and neglect this arguable most crucial phase in the data publishing process. As a second main contribution, we aim to close this gap by focussing our quality assessment approach on RDB2RDF transformations. Consequently, the approach we present is able to detect problems earlier and fix them more easily than the current state of the art on a significant subset of published RDF data. Moreover, by tackling problems at their source, the methods are more efficient and some of the metrics we propose can be applied to very large datasets on which general purpose methods fail.

Overall, in this article, we make the following contributions to the state of the art:

- Definition of a quality assessment methodology focussing on RDB2RDF mappings based on formalizations of the mapping model.
- To the best of our knowledge, providing the first formal approach to RDF quality assessment substantiated by the provision of 43 metric definitions.
- An evaluation of the approach on three real datasets using an open-source reference implementation of the approach.

We first treat related work in Section 2 and present the basic ideas of our approach in Section 3. This is followed by a formal description of our methodology in Section 4. Afterwards we discuss which quality dimensions need consideration in an RDB2RDF quality assessment in Section 5. The formal introduction of selected metrics is given in Section 6. After a brief description of our software prototype in Section 7 we present the results of real world assessments of three different RDB2RDF mapping projects. Finally, we conclude in Section 8.

2. Related Work

Data or information quality is not just considered since the presence of digital information or database systems. First publications on the subject go back to the 1940s when Juran coined the popular definition of quality being *'fitness for use'* [14]. However concrete investigations on data quality models and quality assessment methodologies were mainly published in the last three decades. Whereas [24] proposes a model that has similarities to Shannon's idea of a noisy channel and regards data quality as the extent to which data is *captured* without errors, [31] focuses more on the *representation* of real world data in information systems. Here, data quality deficiencies are described as incomplete or ambiguous representations, or cases where information systems contain data that does not represent real world information or entities. Other models [23,27,21] examine data quality from a more process-oriented perspective regarding data as information products that are created in many steps, each possibly influencing the data's quality.

Based on these fundamental ideas different quality evaluation approaches and systems where developed in the database area [5,26,32,1]. Most of them refer to one of the proposed data quality models to derive a certain data quality score quantifying the data base's quality.

Many of the metrics introduced for the data quality assessment in the database domain were later adapted or extended for the evaluation of Semantic Web and Linked Data [34]. Although the metrics collected and proposed there concern Linked Data quality and are introduced rather informally we took this work as one starting point to derive adapted metrics for the RDB2RDF case. Apart from this, new approaches and methodologies were developed tailored to the characteristics of *semantic* data distributed in a data web. Actual implementations are for example the *SWIQA* tool [7] and *RDFUnit* [16] which utilize SPARQL queries to find data errors in RDF datasets. Although one can find metrics in our metric definitions that are similar to some of their quality checks covering parts of the OWL and RDFS semantics the respective approaches differ. Whereas [7] and [16] derive actual quality checks from the dataset under assessment an RDB2RDF quality assessment method should take the RDB2RDF transformation process into account. A further direction is the *'user-driven quality evaluation'* methodology [33] implemented in the *TripleCheckMate* [17] tool which allows crowd-sourced data quality evaluations of RDF data. Even though crowdsourcing the quality evaluation of RDB2RDF mappings would work in general we doubt that there is a sufficiently large group of RDB2RDF experts available to make this work in practice.

Recently, efforts were made to informally describe concrete data problems in the Semantic Web and set up data quality metrics [9,12,8,13]. We also consid-

ered these ideas and could derive some metrics for our formal framework concerning for example data quality aspects like interpretability and representational conciseness. Considering not just single data quality aspects but covering a wide range of diverse quality issues also lead to more flexible and holistic architectures like the *Sieve* [22] or ODCleanStore [15] frameworks which allow to add and run many different data quality metrics. However, their respective processing pipelines do not meet the requirements of a RDB2RDF quality assessment.

Only few of these considerations were already applied to the RDB2RDF domain, yet. The main aspects investigated cover semantic issues like how to accurately map foreign key relationships [18] or other database constraints [20] to RDF. The corresponding discussions do not include any metrics, though.

So, while there are metrics dedicated to other domains that could be re-used for the RDB2RDF case there is (to the best of our knowledge) no tool support guiding the generation of RDB2RDF mappings or evaluating their quality. Moreover, we could not find an existing method or methodology applicable in the RDB2RDF case.

3. Approach

The assessment approach proposed in this article is based on the mapping model shown in Figure 1. In the depicted workflow RDF data is generated based on quad patterns which can contain variables. These variables generate RDF terms based on constants and relational data. Such RDF terms can be URIs, blank nodes and typed or plain literals, each generated by a corresponding term constructor. The underlying relational data can be referenced via custom SQL queries, existing tables or views defined in the database system. Mapping configurations complying with this model are in the following referred to as *(RDB2RDF) view definitions*.

The main idea of our assessment approach is to consider different scopes a metric can be defined on. Thus, to determine an actual quality score, a metric can use all the context information provided within the metric's scope. This means that, e.g. given a certain metric that assesses whether a plain literal's language tag was validly constructed in accordance with the corresponding standard BCP 47 *'Tags for Identifying Languages'*[3], only node scope is required. Hence, the metric successively gets all RDF nodes of a dataset, i.e. resources and literals, as input and may compute quality scores and metadata to be written to a configured sink implementation.

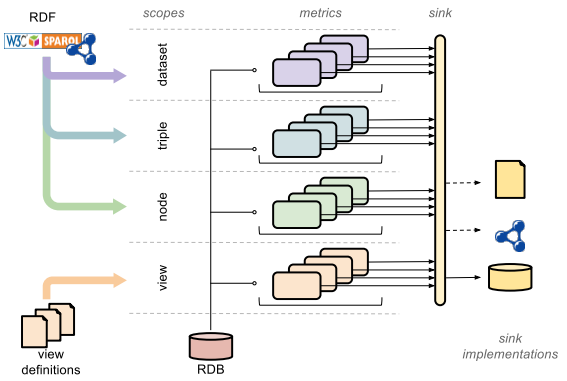

Fig. 2. Conceptual overview of the proposed assessment framework

Since all metrics have the option to access the underlying relational database, our assessment approach considers the actual input of an RDB2RDF mapping, the mapping definitions and the generated output. Nonetheless, the framework is not intended to do any quality evaluations of the relational source data, since this is a separate research field not covered here. In fact, it should give feedback and hints concerning different aspects of data quality that can be influenced by RDB2RDF mappings. One design decision was to provide metrics, which each cover single issues of data quality. Accordingly, we do not take metric interdependencies into consideration which allows a modular design which is highly configurable. However, this also means that our framework is not tailored for a specific use case and not all metrics will be suitable for all application scenarios. Of course, particular implementations of the approach may take interdependencies of metrics into account for improving efficiency as long as they adhere to the formal specification of each metric involved. The general assessment framework is depicted in Figure 2 which gives a conceptual overview.

4. Methodology

To introduce our methodology, a formal terminology needs to be defined first. In the following defini-

[3] http://tools.ietf.org/rfc/bcp/bcp47.txt

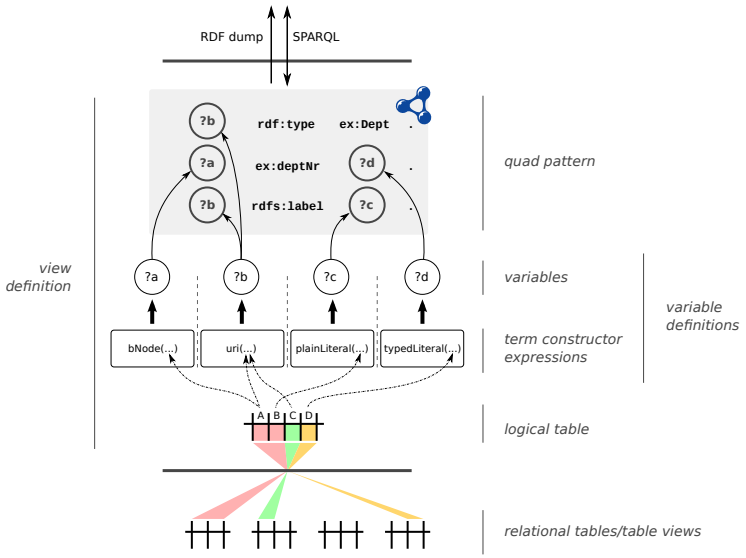

Fig. 1. Conceptual model for RDB2RDF mappings

tions the nth entry of a tuple k is represented by $k_{[n]}$ and $\mathbb{P}$ denotes the power set function.

A *relational database* comprises the set of *tables* $RDB = \delta_1, \delta_2, \ldots, \delta_n$ with each δ_i $(1 \leq i \leq n)$ being a set of tuples. Moreover every δ_i has a set of columns $\Gamma_i = \{\gamma_{i1}, \gamma_{i2}, \ldots, \gamma_{im}\}$, with each column γ_{ij} $(1 \leq j \leq m)$ being a set of values belonging to a certain domain $dom(\gamma_{ij})$. Accordingly a table δ_i can be described as $\delta_i \subseteq (dom(\gamma_{i1}) \times dom(\gamma_{i2}) \times \ldots \times dom(\gamma_{im}))$.

Besides the sets $\mathcal{R}$ of RDF resources, $\mathcal{L}$ of RDF literals and $\mathcal{Q}$ of query variables, one basic term, conceptually introduced in Figure 1, is the notion of a view definition.

Definition 4.1 *A **view definition** $v = (W, TC, \delta)$ is a transformation description that, applied to its logical table δ, generates RDF data. The actual RDF statements are built using a quad pattern $W = \{w_1, w_2, \ldots, w_l\}$, which may contain several quads w_k $(1 \leq k \leq l)$, defined as in [10]. This means that quads are quadruples having a graph (first), subject (second), predicate (third) and object position (fourth), each possibly assigned with a variable $q \in \mathcal{Q}$.[4] Variables are utilized to introduce data stemming from the logical table δ. For each relational data entry a variable q is instantiated to an RDF term based on an associated term constructor $tc_q \in TC$. Such a term constructor has a certain term type to generate either RDF resources, blank nodes, or RDF literals, i.e. $term\_type(tc_q) \in \{\texttt{uri}, \texttt{bNode}, \texttt{plainLiteral}, \texttt{typedLiteral}\}$. Term constructors build RDF terms combining relational data with constants like URI prefixes, data types etc. To address the desired relational data a term constructor references a set of relational columns $cols(tc_q)$ of δ.*

Accordingly, $\mathcal{V}$ refers to the set of all view definitions, adhering to this specification. To access the subject, predicate or object of a quad w the functions $s(w) = w_{[2]}$, $p(w) = w_{[3]}$ and $o(w) = w_{[4]}$ are introduced, respectively. Moreover, these functions can also be used to retrieve all subjects of a *set* of quads, e.g. $s(W) = \bigcup_{w \in W} s(w)$. Besides this, the more readable helper functions $rel\_table(v) = v_{[3]}$ and $quads(v) = v_{[1]}$ are defined for a view definition $v \in \mathcal{V}$. Furthermore, the expression $term\_constr(q)$ yields the term constructor tc_q of a variable q.

Considering the example in Figure 1, W would contain the quads

- '`?b rdf:type ex:Dept`',
- '`?a ex:deptNr ?d`' and
- '`?b rdfs:label ?c`'

with `?a`, `?b`, `?c`, `?d` $\in \mathcal{Q}$. Moreover TC contains the term constructors

- '`bNode(A)`' (with `A` serving as blank node identifier) ,

[4] Please note that graphs are not considered here for brevity. Nonetheless, the term *quad* is used here to be able to distinguish between quads in a view definition's quad pattern and quads in an RDF dataset, which are referred to as *triples*.

- 'uri(A,C)' (constructing RDF terms based on the concatenation of the tuples' relational attributes A and C),
- 'plainLiteral(B, 'en')', and
- 'typedLiteral(D, xsd:string)'

with

- *term_constr*(?a) = bNode(A)
- *term_constr*(?b) = uri(A,C)
- *term_constr*(?c) = plainLiteral(B, 'en')
- *term_constr*(?d) = typedLiteral(D, xsd:string)

Note that the datatype argument of the typedLiteral term constructor and the optional language tag argument of the plainLiteral term constructor are not shown in Figure 1 and were added arbitrarily for this example.

Another integral part of our approach is the notion of an RDB2RDF mapping, defined as follows:

Definition 4.2 *Let $\mathcal{T}$ be the set of RDF triples that are valid according to [10] and $\mathcal{D} = \mathbb{P}(\mathcal{T})$ denote the set of RDF datasets. An **RDB2RDF mapping** H is a tuple (V, RDB, D) where*

- *$V \subset \mathcal{V}$ is a finite set of view definitions*
- *RDB is defined as above*
- *$D \in \mathcal{D}$ is the RDF dataset generated when applying all view definitions $v_i \in V$ to the relations in RDB. $D \subset \mathcal{T}$ is a finite set of RDF triples*[5].

Based on this, the conception of a *scope* can be defined:

Definition 4.3 *Given the sets $\mathcal{R}$, $\mathcal{L}$, $\mathcal{Q}$, $\mathcal{T}$, $\mathcal{D}$, $\mathcal{V}$ defined as above and $\mathcal{N} = \mathcal{R} \cup \mathcal{L} \cup \mathcal{Q}$ denoting the set of nodes, i.e. all resources $\mathcal{R}$, literals $\mathcal{L}$ and quad variables $\mathcal{Q}$.*

*The **quality assessment scope** of a piece of data x is a function defined as follows*

$$scope(x) = \begin{cases} S_N & \text{if } x \in \mathcal{N} \\ S_T & \text{if } x \in \mathcal{T} \\ S_D & \text{if } x \in \mathcal{D} \\ S_V & \text{if } x \in \mathbb{P}(\mathcal{V}) \end{cases} \quad (1)$$

with S_N being the node scope*, S_T being the* triple scope*, S_D being the* dataset scope *and S_V being the* view scope.

[5] It has to be noted, that this dataset definition differs from the common definition of a dataset as a set of graphs, that consist of triples [10]. Even though, the formal framework as well as the proposed metrics could also be introduced based on that definition, graphs are not considered here for brevity.

Accordingly, the scope is a categorization of the granularity a certain piece of data has. This is useful since different *'amounts'* of context information can be needed for the assessment. These amounts correspond to the introduced scopes, i.e. they can either be the whole dataset, one triple, one node or a set of view definitions. These scopes also correspond to the possible domains of the functions that do the actual computation of a quality score.

Definition 4.4 *A **mapping quality metric** M is a pair (f, θ) where f is a quality score function and θ is a numerical value representing a threshold. A quality score function f computes a numeric quality score $f(x)$ of a piece of data x. The range of f is [0,1] with 0 reflecting the worst possible quality score and 1 reflecting the highest possible quality score.*

A mapping quality metric $M = (f, \theta)$ can be further classified as follows:

$$M \text{ is called } \begin{cases} \text{node metric} & \text{if } dom(f) = \mathcal{N} \\ \text{triple metric} & \text{if } dom(f) = \mathcal{T} \\ \text{dataset metric} & \text{if } dom(f) = \mathcal{D} \\ \text{view metric} & \text{if } dom(f) = \mathbb{P}(\mathcal{V}) \end{cases}$$

where $dom(\ldots)$ returns the domain of a function.

To initialize an assessment run, a configuration is needed, which is defined as follows:

Definition 4.5 *A **quality assessment configuration** C is a set of mapping quality metrics $\{M_1, M_2, \ldots, M_n\}$ representing all metrics enabled for an assessment, together with their threshold initializations.*

This conceptualization allows enabling and disabling metrics to fit the given assessment needs as well as defining the per metric thresholds. The threshold concept was introduced to reduce the amount of measurement data and to be able to concentrate on cases that are considered critical, as only those quality scores are reported that are worse than the configured threshold. Accordingly, this threshold also distinguishes data that should be considered deficient (score $< \theta$) from data with a sufficient quality (score $\geq \theta$).

Definition 4.6 *A **quality assessment** (H, C, S) is the process of evaluating the quality score function f_i of every metric $M_i \in C$ on a certain RDB2RDF mapping H with*

- *$D \in \mathcal{D}$ being the RDF dataset generated by H*
- *$V \subset \mathcal{V}$ being the view definitions of H .*

– $N = s(D) \cup p(D) \cup o(D)$ *being the set of nodes in* D

The overall assessment result ρ *is defined as*

$$\rho = \bigcup_{M_i \in C} \begin{cases} \bigcup_{n \in N} f_i(n) & \text{if } M_{i[1]}{=}f_i \wedge dom(f_i){=}\mathcal{N} \\ \bigcup_{t \in T} f_i(t) & \text{if } M_{i[1]}{=}f_i \wedge dom(f_i){=}\mathcal{T} \\ f_i(D) & \text{if } M_{i[1]}{=}f_i \wedge dom(f_i){=}\mathcal{D} \\ \bigcup_{v \in V} f_i(v) & \text{if } M_{i[1]}{=}f_i \wedge dom(f_i){=}\mathbb{P}(\mathcal{V}) \end{cases} \tag{2}$$

ρ *is then stored in an assessment sink* S.

Given the quality assessment $A = (H, C, S)$, our proposed methodology can be described with the following steps:

1) Assessment configuration The overall configuration of the assessment comprises three parts: providing access to the set of relations *RDB* of the RDB2RDF mapping H, the selection of metrics to apply together with their thresholds (C) and the configuration of the assessment sink S to write the assessment results to.

2) Automatic assessment run After the configuration, the actual assessment is run, examining the RDB2RDF mapping on the different scopes. The assessment runner feeds

– all dataset metrics with the generated dataset D
– all triple metrics with all triples $t \in D$
– all node metrics with all nodes $n \in N$
– all view metrics with the set of view definitions V defined in H

3) Result analysis After the assessment finished, all assessment results ρ are written to the sink S. Depending on the utilized sink, they can now be further aggregated, visualized or stored to document a temporal quality progress. Since in practice the results can also be enriched with metadata to locate the actual error causes (cf. Section 7) a manual repair phase may follow.

5. Quality Dimensions for RDB2RDF Quality Assessments

A common practice for the evaluation of information or data quality is to measure different *quality dimensions* separately. Following the common perception of data quality as 'fitness for use' [14], this multi-dimensional view allows to select dimensions of importance for quality considerations of a certain usage scenario. To compile such a selection, a common and general categorization is to follow either a theoretical, empirical or intuitive approach (cf. Figure 3) [1]. Whereas a theoretical approach requires a quality model to systematically derive a selection of important dimensions, the empirical method chooses dimensions based on user surveys. An intuitive approach just takes the experiences and intuitions of the person setting up the assessment into account without running a user study.

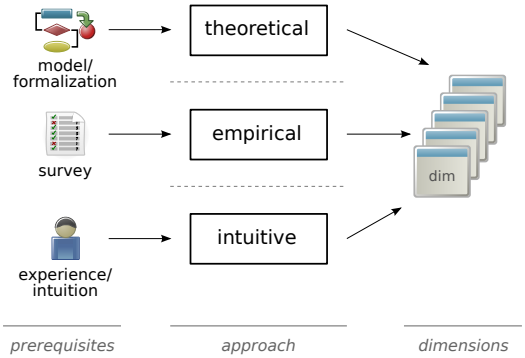

Fig. 3. Approaches to derive quality dimensions to consider for a given domain and their prerequisites

Since the application of the intuitive approach would lack scientific soundness and comprehensibility, it is not considered. Even though taking stakeholder opinions into account would be valid from a scientific point of view, a sufficiently large group of experienced RDB2RDF users and experts is unlikely to be readily available. Thus, empirical results were only regarded indirectly in terms of metrics proposed by other literature sources. The method to obtain quality dimensions, which is considered is the theoretical approach.

To derive quality dimensions of importance for a considered domain on a theoretical basis, a quality model is required. Models viewing data quality as an accurate mapping of the real world to an information system, as proposed e.g. in [31] and [24], are not suitable since in the case of RDB2RDF mappings data of one information system is transformed to another. Moreover, the overall goal of this transformation is not necessarily to make an accurate 'RDF copy' of the relational data. In this article the RDB2RDF method is rather considered as a technique to *model* a domain by means of the Resource Description Framework utilizing given relational data. This means that certain data not contained in the relational database could be added (by means of constants in term constructors) and other

data could be omitted, which is both considered as a quality deficiency in those models. In accordance with this interpretation of RDB2RDF modeling we do not consider the Direct Mapping[6] approach which uses a fixed mapping plan and does not allow custom combinations and the restructuring of the input data.

Since the RDB2RDF mapping can also be regarded as a process with certain steps, a process oriented view can be used to derive a quality model as well. Other than comparing the input and output data, process oriented approaches examine each step in a chain of transformations and modifications. As a consequence a process oriented model provides a more detailed abstraction and is thus more suitable to describe RDB2RDF specific quality concerns. In the following the RDB2RDF mapping process is analyzed with regards to points where data quality degradations may occur. Along with this, quality aspects are considered that are affected by these possible degradations. To divide the RDB2RDF mapping workflow into single steps, the process of SPARQL to SQL rewriting is considered (cf. Figure 4), which can be generalized to a tuple based RDB2RDF transformation. To answer a query received by the SPARQL service (step 1) it has to be parsed and transformed into primitives of the SPARQL algebra (step 2) which does not influence the quality of the response data. Afterwards the query is combined with the mapping definitions and translated into an SQL query (step 3). Since these definitions provide a certain view of the underlying database, this affects quality aspects like *completeness* or *relevance*. Assuming that its actual execution time is neglectable[7], running the generated SQL query (step 4) does not influence the quality. After that, the answer containing the relational result set is retrieved and transformed to RDF (step 5). This transformation may affect *representational* and *syntactic* aspects of the created data, since resource identifiers and literal values are created in accordance with the mapping configuration. Finally, the RDF results are serialized and returned (step 6), where the serialization is a lossless transformation that does not harm the result's data quality.

This shows that the main influencing part within the workflow are the mapping definitions. Nonetheless this does not mean that those mappings really affect all quality aspects. To gather actual dimensions relevant for describing quality issues of RDB2RDF mappings, a shortlisting strategy is applied. Starting with data quality dimensions proposed in a recent and comprehensive survey of quality assessment in Linked Data [34] (cf. Table 1) we evaluated these dimensions with respect to their applicability in the RDB2RDF mapping process using the introduced mapping model and formal foundations (cf. Section 4). The applicability is determined based on two issues. First, a dimension is not applicable if it is not relevant for the RDB2RDF process, i.e. the actual quality score does not depend on the RDF transformation. Moreover, a dimension is also considered not applicable if there are no quality indicators, i.e. it is not possible to actually measure this dimension due to the lack of information needed to do so [3]. In the following, the applicability of the dimensions proposed in [34] is examined. An overview of all dimensions considered relevant for quality assessments of RDB2RDF mappings is given in Table 2.

Table 1
Overview of the dimensions proposed in [34]

Category	Dimension
Accessibility	Availability Licensing Interlinking Security Performance
Intrinsic	Syntactic Validity Semantic Accuracy Consistency Conciseness Completeness
Contextual	Relevancy Trustworthiness Understandability Timeliness
Representational	Representational Conciseness Interoperability Interpretability Versatility

Availability (considered) The availability dimension refers to the extent to which data is *"present, obtainable and ready for use"* [34]. A view definition only indirectly influences the availability of data, namely when URIs are generated that are not dereferenceable. All other aspects of availability are not influenced.

Completeness (considered) Viewing the completeness quality dimension as *"the degree to which in-*

[6] http://www.w3.org/TR/2012/REC-rdb-direct-mapping-20120927/

[7] This assumption was made, since the execution time depends on many different factors that are not within the scope of this article.

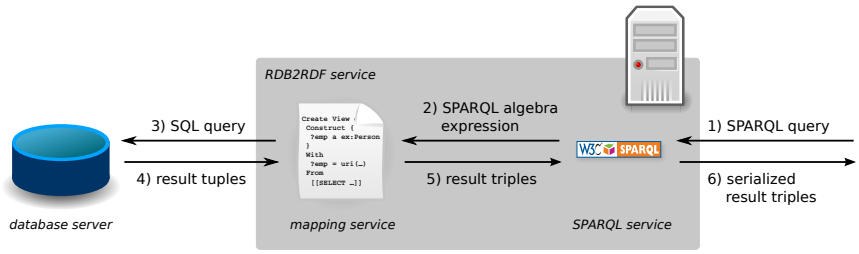

Fig. 4. Steps of the general mapping workflow

Table 2
Overview of the dimensions considered in the RDB2RDF context

Category	Dimension
Accessibility	Availability Interlinking Performance
Intrinsic	Syntactic Validity Semantic Accuracy Consistency Conciseness Completeness
Contextual	Relevancy Understandability
Representational	Representational Conciseness Interoperability Interpretability

formation is not missing" [25] makes it hard to assess it without the provision of a gold standard containing *all* information to compare with. Thus, the weaker completeness notion from the global-as-view approach [30,1] is applied, which refers to the portion of data, that is covered by a view. Since an RDB2RDF mapping can be regarded as an RDF view on a relational database, the completeness term as used here describes how well the underlying database is covered. As there is conceptually no need to map all the data values given in the database to RDF, this completeness aspect is of less importance and should not be seen as a hard quality criterion. Nevertheless, getting feedback of the actual portion of data that is used by a view definition helps finding errors in case the completeness value is much greater or much smaller than expected. Moreover, additional completeness metrics can be introduced, like the interlinking completeness or the completeness with respect to the portion of modeled classes or properties of reused vocabularies.

Conciseness (considered) Conciseness as understood here covers the prevention of any kinds of redundancy on the schema, triple or instance level. This means that multiple RDF properties expressing the same feature and the introduction of duplicate triples should be avoided. Moreover, single database objects should not be mapped to RDF instances multiple times. Such redundancies can be introduced by RDB2RDF mappings and are thus considered.

Consistency (considered) Expressing the *"degree to which the statements of a source's data are conflict-free and no conflicting statements are inferable"* [6], consistency highly depends on the view definitions' term constructors and quad patterns. These can produce datatype inconsistencies or ontology violations and are thus considered.

Interlinking (considered) The importance of providing interlinks to other datasets is reflected in the Linked Data guidelines[8]. Interlinking aspects can be influenced by a view definition's quad pattern and term constructors and are thus subject of the assessment.

Interoperability (considered) Interoperability issues are violations of best practices like term or vocabulary reuse. Since the generated RDF, and thus the degree of reuse, depends on the term constructors and quad pattern defined in a view definition, this dimension is considered in the quality assessment.

Interpretability (considered) This quality dimension covers *"whether information is represented using an appropriate notation"* [34] and depends on generated resource identifiers or literal representations as well as certain quad pattern constructs and thus has to be considered.

Licensing (not considered) The licensing quality dimension is defined as *"degree to which the provided data can be used with own applications"* [6]. Since the terms of usage are already determined by the license used for the relational data, in most cases RDB2RDF tools are not able to influence whether data are *open*

[8] http://www.w3.org/DesignIssues/LinkedData.html

or *restricted*. Only in rare cases where relational data is provided under a very permissive license, it may be republished under more restricted terms of usage by RDB2RDF tools. Apart from this, there is no standardized way of retrieving the actual license information from relational databases. Usually, such licensing meta information is part of the actual relational data to be mapped to RDF. Since there is also no way to detect licensing metadata in a relational database automatically, it can neither be measured, whether the data contained is open or restricted, nor can be determined if there is any licensing information that could have been provided as RDF data. Thus, the licensing dimension is not considered in the quality assessment.

Performance (considered) The mapping process as introduced comprises different services influencing the overall performance. Besides the actual query rewriting engine there is also the relational database with its search and indexing strategies, the actual RDF generation and serialization, as well as network bandwidth and latency when transmitting the query results to the client. The only point where RDB2RDF mapping definitions may influence the performance negatively, is when they contain inefficient SQL queries that define logical tables to map to RDF. This issue is not evaluated since the query optimization topic is already covered widely in the literature. Moreover to optimize a query in an RDB2RDF mapping definition, also database details like existing indexes or the underlying storage architecture have to be taken into account, which may not be accessible to the mapping author.

A further performance aspect, discussed controversially[9] and examined in different sources [6,34], is the usage of hash URIs. In the data quality literature, they are usually considered as bad practice as far as performance is concerned, since in case of accessing a Web resource via a hash URI, the whole document has to be retrieved, even though only a fraction of it was requested. Although the usage of hash URIs has no influence on the performance of the RDB2RDF mapping workflow it is evaluated in the quality assessment to be able to give feedback that a bad practice is applied that may harm the performance *in general*.

[9] See `http://www.w3.org/wiki/HashVsSlash` for a further discussion

Relevancy (considered) Even though there are models to compute the relevancy of a document with regards to a certain topic or keywords [2], it is not trivial to calculate if certain data values are relevant or not. Moreover, since relevancy refers to a certain task and user [25], there is no easy way to determine relevant data in general. The only issues that can be measured are coverage concerns, i.e. how detailed a dataset is or how many resources are described. Relevancy is thus considered in the quality assessment.

Representational Conciseness (considered) Representational conciseness in the Semantic Web context mainly refers to issues of URI design and the usage of certain features of RDF that are considered as deprecated or bad practice [13]. These depend on the term construction and quad design of a view definition and are thus evaluated in the quality assessment.

Security (not considered) Security as a quality dimension mainly covers access control and features to detect unauthorized alteration of data [34]. Since, to the best of our knowledge, current RDB2RDF mapping languages and tools do not provide any means to tackle access control and data integrity, the security dimension is not regarded.

Semantic Accuracy (considered) The semantic accuracy of data generated by RDB2RDF mappings comprises the accurate modeling of the semantics of the relational schema and the relational data. Since there is no explicit semantic description that could be used for a quality assessment, semantically inaccurate *data* can not be detected. Nonetheless, if there are any constraints encoded in the relational *schema*, it can be checked whether these are accurately modeled in the RDF domain. The semantic accuracy dimension is thus evaluated in the quality assessment.

Syntactic Validity (considered) The syntactic validity refers to the correct representation and syntax conformance [34]. Since such syntactical aspects highly depend on the actual usage of the term constructors, the syntactic validity dimension has to be covered in the quality assessment.

Timeliness (not considered) Since current RDB2RDF mapping languages provide no means to influence the *"extent to which data are sufficiently up-to-date for a task"* [25], the timeliness quality dimension is not considered. Moreover, SPARQL to SQL rewriters are capable of transforming relational data to RDF on-the-fly which makes the impact on time dependent aspects neglectable.

Trustworthiness (not considered) Trustworthiness, *"the degree to which the information is accepted to be correct, true, real and credible"* [34], primarily depends on the relation between the data's authors and its users. Since it is not the task of an RDB2RDF mapping tool to keep track of data authors and users, trustworthiness is not included in the quality assessment. Moreover, besides the fact that the authorship of the relational data is usually not evaluated, data from different authors may be mixed up in one single resource or statement, making a trust analysis unfeasible.

Understandability (considered) Understandability refers to the ease of use of data by an information consumer. This ease of use is mainly achieved by a user-friendly URI design and supporting metadata. Since these aspects can be modeled in view definitions, this dimension is evaluated in the quality assessment.

Versatility (not considered) Versatility means the versatility of the supported RDF serializations and versatility with regards to internationalized representations of the data values [34]. The former aspect is usually handled by the RDB2RDF tools, independent of the mapping definitions and the created RDF output and thus does not reflect quality issues of the actual mapping process. Whether internationalized versions of the data exist, depends on the relational data to map and is therefore not subject of the RDB2RDF quality assessment.

6. Metrics for RDB2RDF Quality Assessments

In this section we want to illustrate the utilization of our formal framework to define actual metrics. We therefore selected 7 of our 43 proposed metrics covering diverse use cases to show how the different parts of our formal foundation interact. The metrics introduced in the following provide concrete functions to calculate quality scores for the completeness, consistency, semantic accuracy, interlinking, interpretability and relevancy dimensions. An overview of all our metrics and their corresponding quality dimensions is given in Table 3.

To avoid overly complex mathematical expressions we will sometimes only sketch the quality score function of an introduced metric. In these cases the function's scope, i.e. its domain differs such that for example a function f having the dataset scope might then be presented with a triple input for brevity. Such score functions are then marked with a hat, e.g. $\hat{f}$.

The first two metrics are adaptions of completeness evaluations proposed in [34]. The Population Completeness metric measures the ratio between the number of RDF instances introduced by an RDB2RDF mapping and the objects (i.e. rows, not expressing m:n relations) defined in the underlying relational database.

Metric 1 (Population Completeness) *The metric measuring the ratio between RDF instances and objects of the relational database is a dataset metric. To get the number of database objects of a relation $\delta \in RDB$ with the attributes $\{\gamma_1, \gamma_2, \ldots, \gamma_n\}$, δ's* relational object cardinality $|\delta|_{rel\_obj}$ *is used:*

$$|\delta|_{rel\_obj} = \left|\pi_{\gamma_{p_1},\gamma_{p_2},\ldots,\gamma_{p_m}}(\delta)\right| \tag{3}$$

with $\Gamma_{pk} = \gamma_{p_1}, \gamma_{p_2}, \ldots, \gamma_{p_m}$ representing the (not necessarily compound) primary key of the relation δ and $\pi_{\gamma_j,\gamma_k,\ldots,\gamma_l}(\delta)$ being the projection of δ to its attributes $\gamma_j, \gamma_k, \ldots, \gamma_l$. The cardinality expression of the projection of δ represents the tuple count with duplicate elimination. To avoid counting m:n relations as database objects on its own, a further restriction must hold. Given all referencing foreign key attributes $\Gamma_{fk} = \{\gamma_{f_1}, \gamma_{f_2}, \ldots, \gamma_{f_s}\}$ of δ, the following statement must be true for Equation 3:

$$\Gamma_{fk} \neq \Gamma_{pk}$$

This means that tuples of δ are not counted, if the primary and the foreign key are the same, as in pure m:n relations.

The instance cardinality $|D|_{inst}$ *of a dataset $D \in \mathcal{D}$ is defined as*

$$|D|_{inst} = \left| \left\{ r \;\middle|\; \begin{array}{l} t \in D \wedge r = s(t) \wedge \\ r \notin (\texttt{rdfs:Class} \sqcup \texttt{owl:Class}) \end{array} \right\} \cup \left\{ r \;\middle|\; \begin{array}{l} t \in D \wedge r = o(t) \wedge \\ r \notin (\texttt{rdfs:Class} \sqcup \texttt{owl:Class}) \wedge \\ r \notin \mathcal{L} \wedge p(t) \neq \texttt{owl:sameAs} \end{array} \right\} \right| \tag{4}$$

Thus $|D|_{inst}$ counts all resources not being an `rdfs:Class` *or* `owl:Class`*, whereas objects of* `owl:sameAs` *statements are omitted, to avoid counting resources multiple times that are explicitly stated to be the same. Accordingly the Population Completeness quality score function $f_1 : \mathcal{D} \rightarrow \mathbb{R}$ is given as*

$$f_1(D) = min\left(1, \frac{|D|_{inst}}{\sum\limits_{\delta \in RDB} |\delta|_{rel\_obj}}\right) \tag{5}$$

Table 3
Overview of all metrics and their corresponding quality dimensions

Dimension	Metric	Description
Availability	Dereferenceable URIs	verifies whether URIs can be dereferenced
Completeness	Schema Completeness	computes how many of the available relational columns were referenced in the RDB2RDF mapping
	Population Completeness	computes how many of the database entries were used to generate RDF data
	Property Completeness	computes how many of the possible relational column values were used as property values
	Interlinking Completeness	computes how many of the introduced statements refer to external URIs
	Vocabulary Class Completeness	computes how many of the classes of a reused vocabulary are actually in use
	Vocabulary Property Completeness	computes how many of the properties of a reused vocabulary are actually in use
Conciseness	Intensional Conciseness	verifies whether different RDF predicates describe the same relational attribute
	Extensional Conciseness	verifies whether different RDF resources stem from the same database object or artifact
	No Duplicate Statements	verifies whether an RDB2RDF dump might create duplicate triples
Consistency	Basic Ontology Conformance	verifies whether the generated data has ontological contradictions like invalid datatypes or ranges
	Homogeneous Datatypes	verifies whether there are modeling flaws that introduce different datatypes for one single predicate
	No Deprecated Classes or Properties	verifies whether deprecated classes or properties were introduced by an RDB2RDF mapping
	No Bogus Inverse-functional Properties	verifies whether bogus inverse-functional properties (as reported in [12]) were introduced
	No Ontology Hijacking	verifies whether (conflicting) re-definitions of parts of reused external ontologies were introduced
	No Ambiguous Mappings	verifies whether different database objects were mapped to the same RDF resource
	No Resource Name Clashes	verifies whether the same URIs were introduced identifying conflicting entities, e.g. classes and individuals, classes and properties etc.
	Consistent Foreign Key Resource Identifiers	verifies whether foreign key identifiers and the actual keys they refer to are mapped to the same URIs
Interlinking	External Same-as Links	computes how well the generated RDF dataset is interlinked with other datasets via `owl:sameAs` links
Interoperability	Term Reuse	computes how many of the generated RDF terms refer to external reused URIs
	Vocabulary Reuse	computes how many of the generated vocabulary elements stem from external reused vocabularies
Interpretability	Typed Resources	verifies whether resources are typed properly
	OWL Ontology Declarations	verifies whether resources are related to any ontological structures i.e. whether they are e.g. a subclass, the inverse of a property etc.
	Avoid Blank Nodes	verifies whether blank nodes are generated (as they are considered bad practice [13])
	Correct Collection Use	verifies whether generated RDF collections are free of errors
	Correct Container Use	verifies whether generated RDF container structures are free of errors
	Correct Reification Use	verifies whether generated RDF reification statements are free of errors
Performance	No Hash URIs	verifies whether hash URIs were introduced since they may have an impact the retrieval performance when dereferencing RDF resources

Table 3
Overview of all metrics and their corresponding quality dimensions *(cont.)*

Relevancy	Amount of Triples	computes the amount of triples
	Coverage (Detail)	computes how detailed the descriptions of resources in the RDF data are (i.e. the number of different properties described)
	Coverage (Scope)	computes how many different resources are covered by the generated RDF data
Representational Conciseness	Short URIs	verifies whether overly long URIs were generated by the mapping definition
	No Prolix Features	verifies whether RDF features were introduced which are considered bad practice [13]
	Query Parameter-Free URIs	verifies whether URIs with query parameters were introduced which are considered bad practice [13]
Semantic Accuracy	Preserved NOT NULL Constraints	verifies whether RDF predicates are introduced that lack cardinality constraints but could be restricted to a *minimum cardinality* since they stem from a relational attribute declared to be not null
	Preserved Functional Attributes	verifies whether RDF predicates are introduced that are not declared to be functional but describe usual relational attributes which are functional by definition
	Preserved Foreign Key Constraints	verifies whether one relational entry refers to another entry by means of a foreign key relation and both are mapped to RDF but their *foreign key link* is not expressed
Syntactic Validity	Datatype-compatible Literals	verifies whether generated literal values are compatible with their explicitly declared datatypes
	Valid Language Tags	verifies whether generated language tags comply with the BCP 47 ('Tags for Identifying Languages') standard
Understandability	Labeled Resources	verifies whether resources are labeled
	Sounding URIs	verifies whether resources have a sounding and thus easily memorable URI
	HTTP URIs	verifies whether resources have identifiers that can be used as Web URLs
	Dataset Metadata	verifies whether dataset metadata is provided

The second completeness metric covers the completeness w.r.t. the number of columns and thus the number of possible attributes that could be mapped to RDF. The actual completeness score defined in the following metric expresses the ratio of columns available in the relational tables and all columns referenced in the view definitions of the RDB2RDF mapping.

Metric 2 (Schema Completeness) *The metric assessing the ratio between the number of relational columns referenced in the RDB2RDF mapping and the number of columns that could be referenced, is a view metric. To evaluate the schema completeness, for a given relation $\delta \in RDB$ with the attributes $\{\gamma_1, \gamma_2, \ldots, \gamma_n\}$, δ's* column cardinality $|\delta|_{col} = n$ *is defined as the number of columns in δ. Introducing the* referenced column cardinality $|V|_{ref\_col}$ *of a set of view definitions $V \subset \mathcal{V}$ as*

$$|V|_{ref\_col} = \left| \bigcup_{v_i \in V} \left\{ \gamma' \;\middle|\; \begin{array}{l} q \in \left\{ \begin{array}{l} s(quads(v_i)) \cup \\ p(quads(v_i)) \cup \\ o(quads(v_i)) \end{array} \right\} \cap Q \wedge \\ \gamma' \in cols(term\_constr(q)) \end{array} \right\} \right| \quad (6)$$

the Schema Completeness quality score function $f_2 : \mathbb{P}(\mathcal{V}) \rightarrow \mathbb{R}$ is computed as follows:

$$f_2(V) = \frac{|V|_{ref\_col}}{\sum_{\delta \in RDB} |\delta|_{col}} \quad (7)$$

Again it has to be noted that these two metrics should not be regarded as 'hard' measures for data quality but should give some feedback on how well a given view definition makes use of the data contained in a relational database.

Another issue that might not be a real error but a hint at erroneous mappings is the consistent usage of datatypes. In the RDB2RDF context inhomogeneous datatypes may occur if different view definitions use the same property but apply different, maybe even conflicting types to the properties' values. This might be an indicator of a typo or copy-and-paste error. The corresponding metric is given as follows:

Metric 3 (Homogeneous Datatypes) *The metric assessing the homogeneity of the datatypes of property*

values, is a dataset metric. Given a dataset $D \in \mathcal{D}$ the following set is created to track occurrences of properties and their value types:

$$M = \bigcup_{t \in D} \left\{ (r, type) \middle| \begin{matrix} r = p(t) \wedge o(t) \in \mathcal{L} \wedge \\ o(t) \text{ is of type 'type'} \end{matrix} \right\} \quad (8)$$

The function $\hat{f}_3 : \mathcal{R} \to \mathbb{R}$ determining the quality score of a predicate $r \in \mathcal{R}$ is then defined as follows:

$$\hat{f}_3(r) = \begin{cases} 0 \text{ if } \left| \left\{ (r_M, type) \middle| \begin{matrix} (r_M, type) \in M \wedge \\ r = r_M \end{matrix} \right\} \right| > 1 \\ 1 \text{ otherwise} \end{cases} \quad (9)$$

The Preserved Functional Attributes metric, described in the following, checks whether RDF properties derived from functional relational attributes are declared appropriately. The idea behind this metric is that whenever RDF resources or literals are generated using functional relational attributes from the underlying database, their characteristic of being functional should be reflected somehow in RDF. Considering for example a single row of a relational table all non-primary key fields of this row are functional relational attributes. These functionally depend on the (compound) primary key. Hence, whenever triples are introduced with a subject being generated from primary key values and an object stemming from a functionally dependent attribute, this information of functionality should be preserved in the RDF data. This is usually done assigning the `owl:FunctionalProperty` type to the respecting RDF property.

Metric 4 (Preserved Functional Attributes) *The metric assessing the preservation of relational attributes' characteristics of being functional is a view metric. Given a set of view definitions $V \subset \mathcal{V}$, the set of quad object variables, whose term constructors refer to functional columns of the underlying relational table, can be defined as follows:*

$$Q_{func} = \bigcup_{v_i \in V} \left\{ n_o \middle| \begin{matrix} w \in quads(v_i) \wedge \\ n_s = s(w) \wedge n_o = o(w) \wedge \\ n_s \in \mathcal{Q} \wedge n_o \in \mathcal{Q} \wedge \\ \exists \Gamma \left(\begin{matrix} tc = term\_constr(n_s) \wedge \\ \Gamma \subseteq cols(tc) \wedge \\ \Gamma \text{ is primary key} \\ \text{of the underlying} \\ \text{relational table} \end{matrix} \right) \end{matrix} \right\} \quad (10)$$

Q_{func} then contains all quad variables on object position that should be declared functional via the appropriate type of the corresponding property. The function $\hat{f}_4 : \mathcal{N} \times \mathcal{N} \times \mathcal{N} \to \mathbb{R}$ assigning a quality score to a quad $w \in quads(v_i)$ of a view definition $v_i \in V$ is given as

$$\hat{f}_4(w) = \begin{cases} o(w) \in Q_{func} \wedge \\ 0 \text{ if } \nexists w_f \left(\begin{matrix} w_f \in \bigcup_{v_i \in V} quads(v_i) \wedge \\ s(w_f) = p(w) \wedge \\ p(w_f) = \texttt{rdf:type} \wedge \\ o(w_f) = \\ \texttt{owl:FunctionalProperty} \end{matrix} \right) \\ 1 \text{ otherwise} \end{cases} \quad (11)$$

To build a *web* of data, the linking between different datasets is of crucial importance. This is also reflected in the Linked Data principles[10] and guidelines [11]. Thus, to provide a high quality RDB2RDF mapping, an adequate portion of interlinks should be contained. This is assessed by the following metric.

Metric 5 (External Same-as Links) *The metric assessing the amount of statements expressing that a local and an external identifier refer to the same resource, is a dataset metric. Given a dataset $D \in \mathcal{D}$ and the set of local resources R_{local} with*

$$R_{local} = \bigcup_{t \in D} \left\{ r \middle| \begin{matrix} r \in \Big(s(t) \cup p(t) \cup o(t) \Big) \cap \mathcal{R} \wedge \\ \text{the string representation of } r\text{'s} \\ \text{URI starts with a local prefix} \end{matrix} \right\} \quad (12)$$

a triple $t \in D$ is considered as external same-as link *if*

- $s(t) \in R_{local} \wedge p(t) = \texttt{owl:sameAs} \wedge o(t) \notin R_{local}$, *or*
- $s(t) \notin R_{local} \wedge p(t) = \texttt{owl:sameAs} \wedge o(t) \in R_{local}$.

The number of external same-as links of D is expressed with $|D|_{ext\_sa}$. The quality score function $f_5 : \mathcal{D} \to \mathbb{R}$ is defined as

$$f_5(D) = \frac{|D|_{ext\_sa}}{|D|} \quad (13)$$

When modeling RDF data by means of RDB2RDF techniques there are general statistics that might be helpful to get feedback with regards to the current state of the resulting dataset as well as tracking developments over time. Two statistical values that are introduced in the following are the coverage with respect

[10] `http://www.w3.org/DesignIssues/LinkedData.html`

to the level of detail of a dataset and with respect to its scope. As introduced in [6], these characteristics reflect the aim of providing enough properties to describe resources in detail, and of having enough of these resources to cover the considered domain. Accordingly, if a dataset contains only few distinct RDF properties, its coverage with respect to the level of detail is low. On the other hand, if there are actually only few instances described in the dataset the scope coverage is considered to be bad. Thus, both aspects are contradictory in the sense, that a dataset can not have a perfect scope coverage and detail coverage at the same time. Instead, increasing one of them lowers the other one. The corresponding metrics are defined as follows:

Metric 6 (Coverage (Detail)) *The metric assessing the coverage of a dataset with respect to its level of detail is a dataset metric. For a dataset $D \in \mathcal{D}$ this coverage is given as the ratio of the number of properties actually in use*

$$|D|_{prop} = |p(D)| \tag{14}$$

and the number of triples $|D|$.

The quality score function $f_6 : \mathcal{D} \rightarrow \mathbb{R}$ for an input dataset D is defined as follows:

$$f_6(D) = \frac{|D|_{prop}}{|D|} \tag{15}$$

Metric 7 (Coverage (Scope)) *The metric assessing the coverage of a dataset with regards to its scope is a dataset metric. For a dataset $D \in \mathcal{D}$ this coverage is given as the ratio of the number of instances $|D|_{inst}$ (as introduced in Metric 1), and the number of triples $|D|$. The quality score function $f_7 : \mathcal{D} \rightarrow \mathbb{R}$ for an input dataset D is defined as follows:*

$$f_7(D) = \frac{|D|_{inst}}{|D|} \tag{16}$$

7. Implementation and Evaluation

This section covers implementation and application concerns of our assessment approach. After a brief introduction of our software prototype, we discuss the results of practical assessment runs.

The R2RLint tool is the software prototype implementing our proposed methodology and metrics for RDB2RDF mappings defined in the Sparqlification Mapping Language (SML)[11] [29]. The tool is intended to support RDB2RDF mapping authors giving feedback about the data quality of the resulting RDF data and pointing out directions for improvement. To achieve this the framework provides means to not only record a metric's quality score output but also associated metadata that can be helpful to e.g. trace down the actual causes of a bad score value. Therefore we implemented a service called *pinpointer* which, given a certain triple, can return all quads (together with their actual view definitions, term constructors and source relations) that might have caused the triple's generation. Further metadata might be timestamps to track a temporal development of the assessment results. A metric developer is free to add any information available during the assessment run.

R2RLint is written in Java and designed as a command line tool, aligned with the requirements for quality evaluation frameworks [19,3]. Due to the decoupling of the assessment runner, the overall assessment configuration and the actual metrics, the tool allows users to customize the overall assessment by defining which metrics to apply with which thresholds. Even though R2RLint is equipped with 43 metrics, the framework is designed to be easily extensible with own metric implementations. Providing clear interfaces and utilizing technologies of the Spring framework[12], no explicit wiring and deeper insights into the actual framework are necessary.

Our prototype currently lacks complete SQL query parsing and evaluation support which affects five of our metrics. This means that a view definition might not be assessed in case it refers to a logical table defined by an SQL query. All evaluation results that might be influenced by this shortcoming were marked accordingly.

R2RLint is provided as free software[13]. The tool was used for the practical assessment runs, described in the following.

To use our framework in real world assessments we chose three different datasets generated applying RDB2RDF techniques. The first data source under assessment is part of the *LinkedGeoData* [28] project, which is the RDF version of OpenStreetMap[14]. LinkedGeoData provides spatial data stemming from crowd-

[11] Please note that the mapping model introduced in Section 4 is generic enough to also support R2RML.

[12] `http://projects.spring.io/spring-framework/`

[13] `https://github.com/AKSW/R2RLint`

[14] `http://openstreetmap.org`

Table 4
General statistics of the assessed datasets

Dataset	Triples	Distinct Resources	Literal Values
LinkedGeoData	13,726,852	3,726,142	6,200,583
LCC (Eng)	656,704	128,582	149,788
LinkedBrainz	197,399,205	1,048,239	92,183,398

sourced user input covering the entire globe. Since the amount of data is far too much to be assessed as a whole, only a small portion of LinkedGeoData was chosen for evaluation. This portion was created using the OpenStreetMap database snapshot for the smallest of Germany's federal states, Bremen. Even though the RDF dataset of Bremen is just a small portion of the whole dataset provided by the project, it is referred to as LinkedGeoData in the following for brevity. The dataset was chosen as a medium size dataset with RDB2RDF mapping definitions that are expected to have high quality as the dataset is maintained for several years and backed by GeoKnow[15], a research project aiming at connecting heterogeneous spatial data with Semantic Web technologies.

The second dataset that was evaluated is an RDF version of the English part of the *Leipzig Corpora Collection (LCC)* provided by the *Wortschatz* project of the University of Leipzig[16]. The dataset contains per-language statistics about co-occurrences of different words stemming from different corpora, e.g. Wikipedia pages or news sites. It was generated ad-hoc to support the creation of multilingual Linked Open Data applications at the *Multilingual Linked Open Data for Enterprises (MLOD)* conference 2012[17]. Being an ad-hoc attempt, created for a very limited purpose, the mapping is expected to be of poor quality. It does not contain much ontological structures, but merely the core statistics. The RDF data was generated using the 10K version of a tab-separated values (TSV) dump[18] holding statistics of words and stemming from 10,000 sentences of the English Wikipedia. The dataset is referred to as LCC (Eng) in the following.

An overview of the datasets used for the evaluation is given in Table 4.

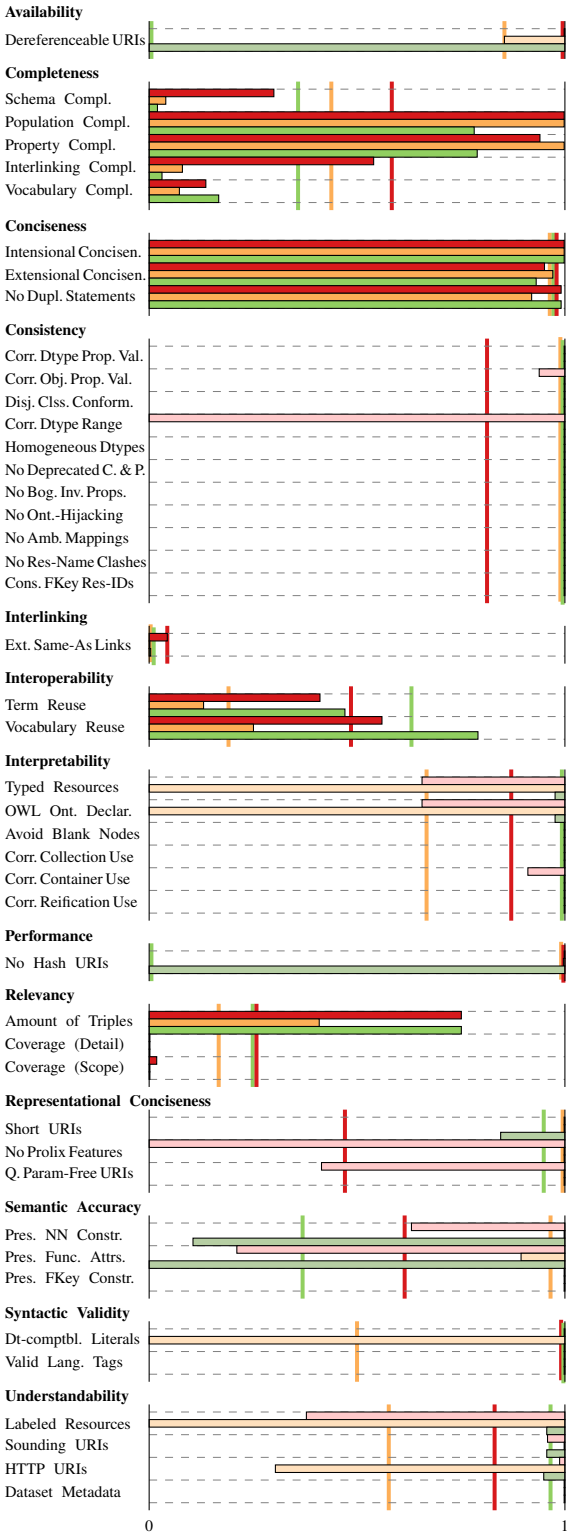

Fig. 5. Comparison of the assessment results of the LinkedGeoData (red), LCC (yellow) and LinkedBrainz (green) datasets by metric.

[15] http://geoknow.eu
[16] http://corpora.uni-leipzig.de
[17] http://sabre2012.infai.org/mlode
[18] http://corpora.uni-leipzig.de/downloads/eng_wikipedia_2010_10K-text.tar.gz
[18] http://download.geofabrik.de/europe/germany/bremen-latest.osm.pbf, retrieved Nov 17, 2013

Table 5

Assessment results aggregated by dimension (see Section 7 for details). For the values marked with * not all metrics of the given dimension were evaluated. Dimensions marked with † contain metrics that might be affected by implementation limitations of the R2RLint prototype described in Section 7.

Dataset	*Completeness*	*Conciseness*†	*Interlinking*	*Interoperability*	*Relevancy*	*Availability*	*Consistency*†	*Interpretability*	*Repr. Conciseness*	*Sem. Accuracy*†	*Synt. Validity*	*Understandability*
LinkedGeoData	0.9071	0.9800	0.0440	0.4850	0.2559	0.00	67.20	1,497.12	4,910.04	7.33	0.00	2,663.26
LCC (Eng)	0.8007	0.9633	0.0000	0.1900	0.1667	17.82	0.06	3,849.38	1.62	0.67	726.58	6,677.89
LinkedBrainz	0.3575	0.9733	0.0010	0.6300	0.2500	121.54	0.00*	88.91	477.73	12.00	0.00	546.77

The last RDB2RDF mapping project under assessment is *LinkedBrainz* which provides SPARQL access to an RDF version of the MusicBrainz database. Initially funded by the non-departmental public body Jisc[19], LinkedBrainz later became part of the EUCLID[20] EU project. LinkedBrainz is now maintained at the British Museum[21]. Accordingly, this dataset is also expected to be of high quality.

We ran our quality assessment framework on a computer with an Intel i5 2.3GHz CPU with four cores and 4 GB of memory. The overall evaluation results are presented in Table 5 and Figure 5. Table 5 shows the assessment outcome grouped by quality dimension. On the left part of the table the average quality scores are listed for the dimensions' metrics that return a continuous quality score and can be compared amongst the datasets. The right part shows the average number of errors per 100,000 triples for the dimensions' 'binary' metrics, i.e. metrics that report 0 if a quality deficiency was detected and 1 otherwise. Figure 5 shows a graphical comparison of the three RDB2RDF mapping projects under assessment. The bars running from left to right show the single metrics' average quality scores with continuous output and the bars in pale colors running from right to left represent the number of errors per 100,000 triples for binary metrics. The vertical bars show the average score and average error counts per 100,000 triples per dimension, respectively. Error counts were scaled to the range [0, 1] per dimension, i.e. the corresponding bars are not comparable amongst different quality dimensions.

The comparison overview graph shows a diverse result supporting the claim that *(data) quality* has many different facets that make it unlikely to generate allover high quality RDB2RDF data. Moreover there are cases where characteristics that are considered bad in one quality dimensions may lead to a better quality in another dimension. This becomes more obvious when considering the LCC (Eng) dataset. Since it does not contain much ontological information and lacks type definitions, its interpretability scores are quite low, as expected. However this missing information also led to quite good results in the consistency dimension since no ontological violations could be found.

In the following, we discuss single evaluation results of the overall assessment run more deeply. Due to space limitations, we only highlight results that show significant characteristics of the datasets under assessment.

With regards to the availability dimension, the dereferenceability of generated URIs was evaluated. The most non-dereferenceable URIs were found in the LinkedBrainz dataset. These mainly comprise Discogs URLs like `http://www.discogs.com/artist/AC%2FDC`. Even though they can all be looked up in a browser, trying to retrieve them via the corresponding Java libraries or curl command line queries without providing a `User-Agent` HTTP header resulted in a response '500 Internal Server Error'. Further dereferenceability issues arose for `owl:sameAs` links to different DBpedia datasets.

The evaluation of the completeness domain showed that the LinkedBrainz and LCC (Eng) datasets have a low interlinking completeness. With the Property Completeness metric a view definition of the LinkedBrainz RDB2RDF mappings could be detected, which

[19] `http://jisc.ac.uk/`
[20] `http://euclid-project.eu`
[21] `http://www.britishmuseum.org/`

does not generate any triples. The different results of the vocabulary completeness metrics show that only very few vocabularies were modeled completely.

Regarding the conciseness dimension, an obvious deficiency with respect to duplicate statements introduced by RDB2RDF mappings could be detected for the LinkedBrainz dataset. The value of 0.04 evaluated for one view definition showed that a lot of duplicate triples were introduced. In fact, this could be traced back to an erroneous mapping, referring to a wrong relational column.

With regards to the consistency dimension, violations could be observed in the LinkedGeoData dataset. Evaluating RDFS and OWL ontology axioms under a closed world and unique name assumption, as motivated in [16] or the research on the Pellet Integrity Constraint Validator[22], these axioms can be interpreted as *constraints* that must hold. LinkedGeoData contained violations of such constraints for different property range axioms and the ranges of object and datatype properties. Besides this, no further violations were found. In some cases, this can be attributed to rather poor ontologies, where not many of such consistency restrictions can be derived from, as in the case of the LCC (Eng) dataset.

Moreover, LinkedGeoData made statements about *external* resources from the `http://sws.geonames.org/` and the DBpedia resource namespaces, which are thus considered to be *bad smells* with regards to ontology hijacking. Actual hijacking *violations* were also found, since the LinkedGeoData dataset contained ontological re-definitions concerning the `foaf:mbox` property. However, it has to be noted that only one of these four statements differs from the original definitions of the FOAF vocabulary.

The assessment results showed that the scores of the metric assessing the usage of `owl:sameAs` links differ considerably. LinkedGeoData is better interlinked than LinkedBrainz, and LCC (Eng) only provides a very small portion of `owl:sameAs` links.

With regards to their interoperability, the LinkedGeoData and the LinkedBrainz datasets clearly outperform LCC (Eng). Whereas LinkedGeoData and LinkedBrainz have a similar score for the term reuse, LinkedBrainz has the more comprehensive vocabulary reuse. For the LCC (Eng) dataset we manually looked for suitable vocabulary candidates to improve its vocabulary reuse score. Using the local names of their URIs, we queried the LODStats[23] [4] website for alternatives to the classes and properties, not being reused. With this strategy, we could find four vocabulary candidates, that could be reused and hence could increase the reuse score significantly.

For the interpretability dimension an obvious quality deficiency was detected with regards to the typing and the provision of an ontological context for classes and properties in the LCC (Eng) dataset. More specifically, for a considerable portion of the resources it is not clear, whether they are instances, classes or properties. As already noted, this further impacts other metrics like those of the consistency dimension. Besides this, since some metrics use the number of *instances* the missing type statements might also influence the results of these metrics.

Besides this, it could be detected that in LinkedBrainz certain resources are not typed. The resources that are explicitly excluded from the type assignments in the RDB2RDF mappings are MusicBrainz release events that are not dated with a year, month *and* day. Nonetheless, since the LinkedBrainz RDB2RDF mappings also generate release event resources, that are just dated with a year or a year and month, this seems to be an error, especially because all other introduced resources are typed.[24] Another significant error pattern was detected for the LinkedGeoData mappings. There, the first container member is declared using the container membership property `rdf:_0` instead of `rdf:_1`.

The only performance aspect considered relevant for the assessment of RDB2RDF mappings was the introduction of local hash URIs. With respect to the view that hash URIs should be avoided, the LinkedBrainz dataset would be of bad quality, since all local URIs are designed to contain the hash sign. Nonetheless, nearly all of them have the fixed fraction part `#_`. Thus, there are usually no two resources sharing a non-fraction part. Accordingly, the argumentation, that hash URIs would harm the performance does not hold in this case.

The assessment of the representational conciseness dimension, also checking whether short and query parameter-free URIs are introduced, showed that only a smaller portion of the very long URIs can be attributed to a bad URI design. Instead many of the URIs considered as violation are overly long since they completely consist of special characters, being percent-encoded. In these cases one single letter had to

[22] `http://clarkparsia.com/pellet/icv/`

[23] `http://stats.lod2.eu`

[24] It has to be noted that this error was fixed by the mapping authors during our evaluation.

be encoded by at least three characters, which led to long URIs. In this regard, resource identifiers based on characters from writing systems not allowed in URIs, have a clear disadvantage. The only exception, where URIs were considerably long by design was given in the RDB2RDF mappings of the LinkedBrainz dataset. There, URIs were generated that hold two UUID[25] strings, each having 36 characters. Besides this the LinkedGeoData mappings make use of a considerable amount of RDF container statements, e.g. to express paths and polygons. Since container statements are considered as 'prolix features' in some literature sources [13] a bad quality score was reported in this respect. Apart from this LinkedGeoData also provides geolocation geometries in the GADM-RDF withinRegion format[26] which holds latitude and longitude values in query parameters.

The metrics assessing the semantic accuracy of RDB2RDF mappings all refer to certain characteristics of the relational schema definitions of the underlying database. Our assessment results showed a considerable number of inaccuracies in this regard. This means, that there was certain semantic information contained in the corresponding relational databases, that was not considered in the RDB2RDF mappings under assessment. This can be attributed to the circumstance that gathering all these (partly implicit) relational constraints is rather cumbersome if it is done by hand.

Analyzing the assessment results revealed 7 deficiencies that were clear mapping *errors*:

- The usage of a wrong relational column in the MusicBrainz mappings uncovered when we examined the huge number of duplicates reported by the *No Duplicate Statements* metric
- The usage of `rdf:_0` as first container membership property found in the LinkedGeoData mappings discovered by the *Correct Container Use* metric
- The assignment of literal values using the object property <`http://linkedgeodata.org/ontology/wheelchair`> in LinkedGeoData found by the *Basic Ontology Conformance* metric
- The assignment of URIs using the datatype property <`http://linkedgeodata.org/ontology/agricultural`> in LinkedGeoData found by the *Basic Ontology conformance* metric
- The wrong range declaration of <`http://geovocab.org/spatial#Feature`> on the property `foaf:mbox` in the LinkedGeoData mappings pointed at by the *No Ontology Hijacking* metric
- The forgotten type definitions of certain resources in the LinkedBrainz mappings discovered by the *Typed Resources* metric
- The invalid typing of date information as `xsd:dateTime` in a mapping definition stemming from the LCC (Eng) project reported by the *Datatype-compatible Literals* metric

These caused more than 850,000 violations in total, but could be fixed with little effort and time. Usually, the actual corrections are simply updating certain parts of term constructor expressions or changing relational queries.

8. Conclusions and Future Work

In this article a methodology for RDB2RDF quality assessments was developed and an overview of dimensions to consider was given. Moreover a formalism was proposed and used to define metrics exemplarily. Besides these formal and conceptual considerations, a software implementation was used to actually run quality assessments on real world datasets. The evaluation of the assessment results showed detailed quality characteristics of RDB2RDF mapping projects allowing targeted updates to improve data quality.

Apart from this, the developed software prototype showed directions for improvements. One issue is that currently the only implemented, practically relevant assessment sink writes the quality scores and the corresponding metadata to a relational database using a quite complex database schema. To improve the exploration options, in future work further assessment sinks will be implemented, e.g. providing results as RDF data cube[27].

Another shortcoming at this time is the lack of SQL parsing support which will be added soon. A further drawback was that the computation of some metrics took impractically long time or was not feasible at all due to memory shortages when using RDFS or OWL reasoning, or running complex local SPARQL queries on bigger datasets. Thus, the No Resource Name Clashes and Basic Ontology Conformance met-

[25] `http://www.opengroup.org/dce/info/draft-leach-uuids-guids-01.txt`

[26] `http://gadm.geovocab.org/`

[27] `http://www.w3.org/TR/vocab-data-cube/`

rics could not be executed on the LinkedBrainz dataset on our assessment setup. This scalability problem occurred mainly during the computation of metrics requiring dataset scope. Hence, one future task will be to put effort into the transformation of dataset metrics into view metrics and using sampled or surrogate data inferred from the relational schema.

The concept of the quality assessment configuration could be extended to not just switch on and of single metrics but to allow to specify the metrics' weights. This could then be used to compute a combined quality score based on these weights, which might give a clearer overview of the quality of a considered RDB2RDF mapping project.

Besides the actual assessment of existing mapping definitions, the prototype could also be extended to make mapping suggestions, which improve the overall quality. These suggestions comprise e.g. mapping rules to accurately model relational schema constraints in RDF, and proposed vocabularies to improve the vocabulary reuse. Thus, a further vision would be to use the R2RLint reference implementation as back-end for an RDB2RDF editing workbench, which interactively guides RDB2RDF mapping authors to optimize the mappings quality.

Generally, we believe to have made a step towards proper formalisation of RDF quality assessment, which plays a crucial role in academic and industrial use of RDF data. While we focused on RDB2RDF mappings, many concepts and metrics are applicable to general RDF quality assessment.

Acknowledgement

This work was supported by a grant from the European Union's 7th Framework Programme provided for the project GeoKnow (GA no. 318159).

References

[1] C. Batini and M. Scannapieco. *Data Quality: Concepts, Methodologies and Techniques*. Data-Centric Systems and Applications. Springer-Verlag, Berlin Heidelberg, 2010.

[2] C. Bizer. *Quality-Driven Information Filtering in the Context of Web-Based Information Systems*. PhD thesis, Freie Universität Berlin, Mar. 2007.

[3] C. Bizer and R. Cyganiak. Quality-driven information filtering using the wiqa policy framework. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(1):1–10, Jan. 2009.

[4] J. Demter, S. Auer, M. Martin, and J. Lehmann. Lodstats—an extensible framework for high-performance dataset analytics. In *Proceedings of the EKAW 2012*, Lecture Notes in Computer Science (LNCS) 7603. Springer, 2012. 29% acceptance rate.

[5] L. P. English. *Improving Data Warehouse and Business Information Quality: Methods for Reducing Costs and Increasing Profits*. John Wiley & Sons, New York, USA, 1999.

[6] A. Flemming. Qualitätsmerkmale von linked data-veröffentlichenden datenquellen. Diploma thesis, Humboldt-Universität zu Berlin, Mar. 2011.

[7] C. Fürber and M. Hepp. Swiqa - a semantic web information quality assessment framework. In V. K. Tuunainen, M. Rossi, and J. Nandhakumar, editors, *19th European Conference on Information Systems, ECIS 2011, Helsinki, Finland, June 9-11, 2011*, 2011.

[8] C. Guéret, P. T. Groth, C. Stadler, and J. Lehmann. Assessing linked data mappings using network measures. In *Proceedings of the 9th Extended Semantic Web Conference*, volume 7295 of *Lecture Notes in Computer Science*, pages 87–102. Springer, 2012.

[9] H. Halpin, P. J. Hayes, J. P. McCusker, D. L. McGuinness, and H. S. Thompson. When owl:sameas isn't the same: An analysis of identity in linked data. In P. F. Patel-Schneider, Y. Pan, P. Hitzler, P. Mika, L. Zhang, J. Z. Pan, I. Horrocks, and B. Glimm, editors, *The Semantic Web – ISWC 2010: 9th International Semantic Web Conference, ISWC 2010, Shanghai, China, November 7-11, 2010, Revised Selected Papers, Part I*, volume 6496 of *Lecture Notes in Computer Science*, pages 305–320, Berlin Heidelberg, 2010. Springer-Verlag.

[10] S. Harris and A. Seaborne, editors. *SPARQL 1.1 Query Language*. World Wide Web Consortium, Mar. 2013. Available at `http://www.w3.org/TR/2013/REC-sparql11-query-20130321/`.

[11] T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Morgan and Claypool, 1st edition, 2011.

[12] A. Hogan, A. Harth, A. Passant, S. Decker, and A. Polleres. Weaving the pedantic web. In C. Bizer, T. Heath, T. Berners-Lee, and M. Hausenblas, editors, *Proceedings of the WWW2010 Workshop on Linked Data on the Web*, volume 628 of *CEUR Workshop Proceedings*, Aachen, Germany, Apr. 2010. Redaktion Sun SITE, Informatik V, RWTH Aachen.

[13] A. Hogan, J. Umbrich, A. Harth, R. Cyganiak, A. Polleres, and S. Decker. An empirical survey of linked data conformance. *Web Semantics: Science, Services and Agents on the World Wide Web*, 14:14–44, July 2012.

[14] J. M. Juran and A. B. Godfrey. *Juran's Quality Handbook*. McGraw-Hill, New York City, United States, 5th edition, 1998.

[15] T. Knap, J. Michelfeit, J. Daniel, P. Jerman, D. Rychnovský, T. Soukup, and M. Nečaský. Odcleanstore: A framework for managing and providing integrated linked data on the web. In X. S. Wang, I. Cruz, A. Delis, and G. Huang, editors, *Web Information Systems Engineering - WISE 2012*, volume 7651 of *Lecture Notes in Computer Science*, pages 815–816, Berlin Heidelberg, Nov. 2012. Springer-Verlag.

[16] D. Kontokostas, P. Westphal, S. Auer, S. Hellmann, J. Lehmann, R. Cornelissen, and A. Zaveri. Test-driven evaluation of linked data quality. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 747–758, Republic and Canton of Geneva, Switzerland, 2014. International World Wide Web Conferences Steering Committee.

[17] D. Kontokostas, A. Zaveri, S. Auer, and J. Lehmann. Triplecheckmate: A tool for crowdsourcing the quality assessment of linked data. In *Proceedings of the 4th Conference on Knowledge Engineering and Semantic Web*, 2013.

[18] G. Lausen. Relational databases in rdf: Keys and foreign keys. In V. Christophides, M. Collard, and C. Gutierrez, editors, *Semantic Web, Ontologies and Databases: VLDB Workshop, SWDB-ODBIS 2007, Vienna, Austria, September 24, 2007, Revised Selected Papers*, volume 5005 of *Lecture Notes in Computer Science*, pages 43–56, Berlin Heidelberg, Sept. 2007. Springer-Verlag.

[19] Y. Lei, V. Uren, and E. Motta. A framework for evaluating semantic metadata. In D. Sleeman and K. Barker, editors, *Proceedings of the 4th International Conference on Knowledge Capture*, pages 135–142, New York, NY, USA, 2007. ACM Press.

[20] D. V. Levshin. Mapping relational databases to the semantic web with original meaning. In D. Karagiannis and Z. Jin, editors, *Knowledge Science, Engineering and Management: Third International Conference, KSEM 2009, Vienna, Austria, November 25-27, 2009. Proceedings*, volume 5914 of *Lecture Notes in Computer Science*, pages 5–16, Berlin Heidelberg, Nov. 2009. Springer-Verlag.

[21] L. Liu and L. N. Chi. Evolutional data quality: A theory-specific view. In C. Fisher and B. Davidson, editors, *Seventh International Conference on Information Quality (IQ 2002)*, pages 292–304, Cambridge, MA, USA, Nov. 2002. MIT Press.

[22] P. N. Mendes, H. Mühleisen, and C. Bizer. Sieve: Linked data quality assessment and fusion. In D. Srivastava and I. Ari, editors, *EDBT-ICDT '12: Proceedings of the 2012 Joint EDBT/ICDT Workshops*, pages 116–123, New York, NY, USA, Mar. 2012. ACM.

[23] A. Motro and I. Rakov. Estimating the quality of databases. In T. Andreasen, H. Christiansen, and H. L. Larsen, editors, *Flexible Query Answering Systems: Third International Conference, FQAS'98 Roskilde, Denmark, May 13–15, 1998 Proceedings*, volume 1495 of *Lecture Notes in Computer Science*, pages 298–307, Berlin Heidelberg, 1998. Springer-Verlag.

[24] A. Parssian, S. Sarkar, and V. S. Jacob. Assessing data quality for information products: Impact of selection, projection, and cartesian product. *Management Science*, 50(7):967–982, July 2004.

[25] L. L. Pipino, Y. W. Lee, and R. Y. Wang. Data quality assessment. *Communications of the ACM*, 45(4):211–218, Apr. 2002.

[26] T. Redman. *Data Quality – The Field Guide*. Digital Press, 2001.

[27] G. Shankaranarayanan, R. Y. Wang, and M. Ziad. Ip-map: Representing the manufacture of an information product. In B. D. Klein and D. F. Rossin, editors, *Fifth Conference on Information Quality (IQ 2000)*, pages 1–16, Cambridge, MA, USA, 2000. MIT Press.

[28] C. Stadler, J. Lehmann, K. Höffner, and S. Auer. Linkedgeodata: A core for a web of spatial open data. *Semantic Web Journal*, 3(4):333–354, 2012.

[29] C. Stadler, J. Unbehauen, P. Westphal, M. A. Sherif, and J. Lehmann. Simplified rdb2rdf mapping. In C. Bizer, T. Heath, S. Auer, and T. Berners-Lee, editors, *8th Workshop on Linked Data on the Web*, 2015.

[30] J. D. Ullman. Information integration using logical views. In F. Afrati and P. Kolaitis, editors, *Database Theory — ICDT '97: 6th International Conference Delphi, Greece, January 8–10, 1997 Proceedings*, volume 1186 of *Lecture Notes in Computer Science*, pages 19–40, Berlin Heidelberg, Jan. 1997. Springer-Verlag.

[31] Y. Wand and R. Y. Wang. Anchoring data quality dimensions in ontological foundations. *Communications of the ACM*, 39(11):86–95, Nov. 1996.

[32] R. Y. Wang, M. Ziad, and Y. W. Lee. *Data Quality*. Kluwer Academic Publishers, Dordrecht, Netherlands, 2001.

[33] A. Zaveri, D. Kontokostas, M. A. Sherif, L. Bühmann, M. Morsey, S. Auer, and J. Lehmann. User-driven quality evaluation of dbpedia. pages 97–104. ACM Press, Sept. 2013.

[34] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann, and S. Auer. Quality assessment for linked data: A survey. *Semantic Web Journal*, 2015.