A Framework for Assessing and Refining the Quality of R2RML mappings

Alex Randles†  
ADAPT Centre  
Trinity College Dublin  
Dublin 2, Ireland   
 alex.randles@adaptcentre.ie

Ademar Crotti Junior  
ADAPT Centre   
Trinity College Dublin  
 Dublin 2, Ireland   
 ademar.crotti@adaptcentre.ie

Declan O’Sullivan  
 ADAPT Centre   
 Trinity College Dublin  
Dublin 2, Ireland   
declan.osullivan@adaptcentre.ie

ABSTRACT

“Uplift” mapping execution applies a set of mapping definitions to transform non-RDF data sources to RDF. This conversion is typically the last step in the uplift mapping process. During the uplift mapping process, mapping definitions are iteratively refined until they conform with the user’s expressed requirements. The W3C standard R2RML is one language which allows for specifying the mappings needed to uplift relational databases to RDF. Even though many approaches have been proposed to assess the quality of the resulting RDF datasets, the root cause of several of those quality violations are found in mappings, and thus are rarely assessed or adjusted. In this paper, we present a framework for assessing and refining the quality of the definitions used to transform non-RDF data to RDF. RDF datasets are continuously being published and quality assurance is a key aspect to the data’s usability. Hence, our proposed framework brings quality assessment procedures earlier into the uplift mapping process itself to ensure that quality issues can be detected and removed prior to dataset generation and publication. This paper also presents an implementation of the proposed quality assessment framework for the R2RML mapping language, which uses the W3C standard Shapes Constraint Language (SHACL). We also provide a demonstration of the proposed framework and its implementation through a walk-through use case.

KEYWORDS

Mapping Process; Mapping Quality; Data Quality; R2RML.

# INTRODUCTION

Linked Data refers to a set of best practices and guidelines for publishing structured data on the Web [1]. A Linked Data dataset is structured information encoded using the Resource Description Framework (RDF) [2], in which resources are identified by and linked with other datasets using URIs.

Oftentimes, RDF datasets are generated from the process of converting non-RDF resources to RDF. This process (termed “uplift”) commonly involves the definition of mappings, which allows one to declaratively express the transformations needed to convert non-RDF source data to RDF [5]. Mappings are also responsible for detaching mapping definitions from the implementations that execute them, facilitating the reproduction of the process responsible for the generation of datasets (when updating datasets, or creating new ones, where mappings - or parts of - can be reused).

Such mappings are commonly expressed through mapping languages. A prominent and popular example is the W3C Recommendation RDB-to-RDF mapping language (R2RML) [7]. Mapping languages themselves, however, do not ensure the quality of the defined mappings nor of the datasets being generated by those mappings. While several quality assessment frameworks have been proposed [8,26], in most cases, these are only focused on the resulting datasets and not on the mappings used to produce them. A violation within the mapping can propagate exponentially within the resulting data. Therefore, identifying violations within mappings will ensure that the resulting dataset does not contain the identified violations [15], while also improving mapping reuse and facilitating mapping maintenance. For instance, by only assessing and refining the generated datasets and not the mappings used to produce them, data producers are at risk of overwriting the refined datasets when a new version is generated or when mappings are being reused. In addition, previous work [15] has shown that a number of quality procedures which are often executed on the resulting datasets can potentially be brought to the start of the mapping process, where such quality assurance procedures would avoid the propagation of violations to all generated datasets.

In this paper, we propose a framework for assessing and refining the quality of mappings used to generate RDF datasets. This is done by extending quality assessment and refinement procedures to also cover earlier steps of the uplift mapping process. Our proposed framework is designed to ensure the quality, of not only the resulting dataset but also to the defined mapping. We argue that this will allow mapping quality violations to be identified and fixed during mapping design-time i.e. prior to the generation of datasets. The goal of our work is to also define what quality information related to the mapping process should be provided, such that it produces higher quality mappings and datasets while also facilitating the maintenance and reuse of the defined mappings. The main contributions of this paper are (i) a quality assessment framework focused on mappings used to generate and publish RDF datasets; (ii) an implementation of the proposed framework, which includes the definition of mapping quality metrics and relies on the W3C Recommendation Shapes Constraints Language (SHACL) [17]; and (iii) a demonstration of the proposed approach through a walk-through use case.

The remainder of this paper is organized as follows. Section 2 describes the R2RML mapping language. Section 3 presents the proposed mapping quality assessment framework. Section 4 discusses quality categories, dimensions and metrics for mappings. Section 5 presents the implementation of our proposed mapping quality framework. Section 6 describes related work. Section 7 concludes the paper and discusses future work.

# R2RML

The RDB to RDF Mapping Language (R2RML) [7] is the W3C Recommendation for the transformation of relational databases to RDF. R2RML mappings consist of one or more triples maps. The structure of a triples map can be described as follows:

**Logical table.** One logical table, which contains a table, view or SQL query, from which the RDF will be generated.

**Subject map.** One subject map, which describe the subjects of the RDF triples to be generated. Subject maps can be defined as IRIs or blank nodes. Subject maps may also be defined as instances of zero or more class types. Finally, subject maps may be associated with named graphs to which the generated triples will assigned to.

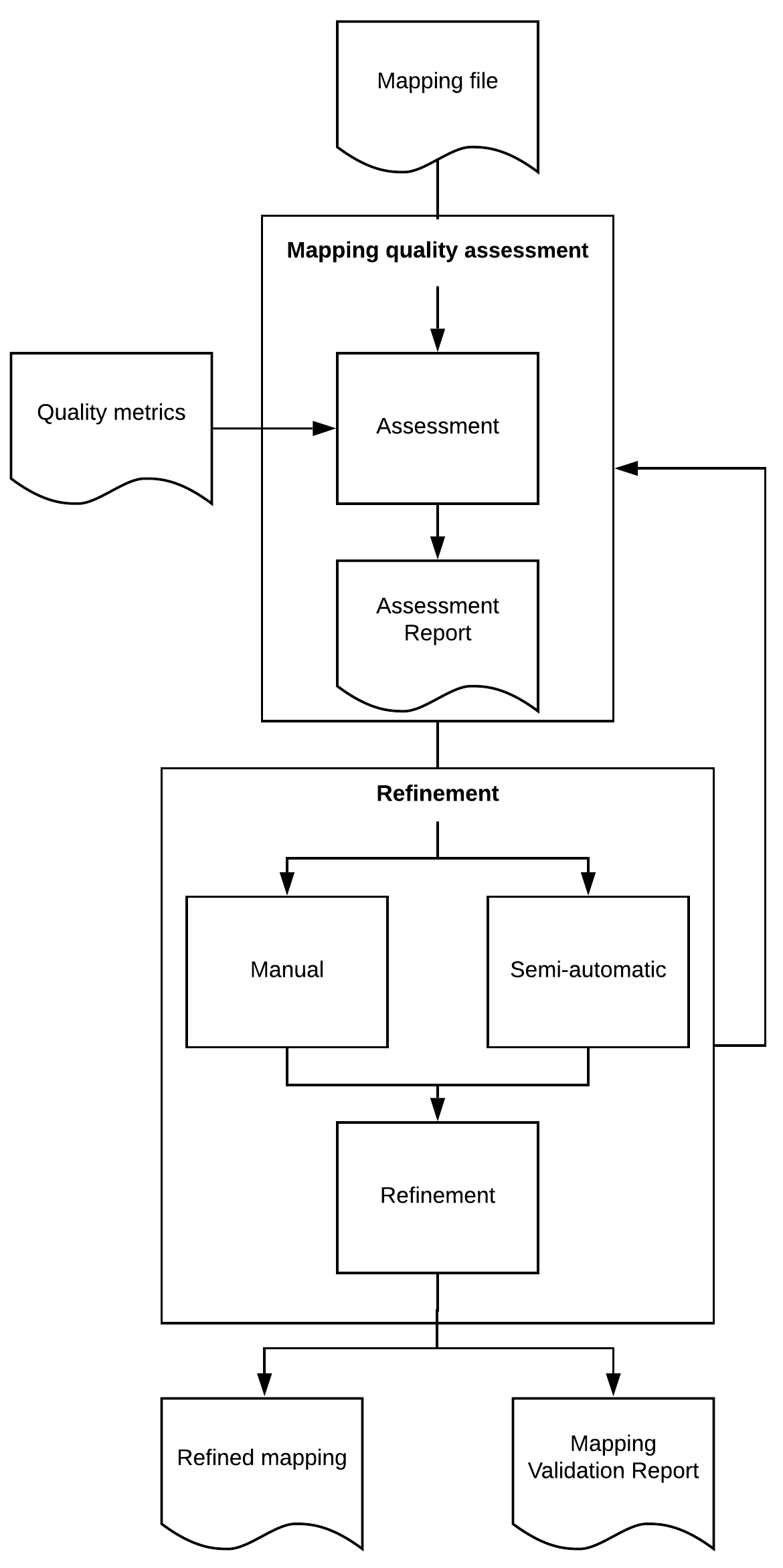
**Predicate Object Map**. Zero or more predicate object maps, which contain the predicate and object of the RDF triples being generated. Predicates are defined using predicate maps, while objects are defined using object maps. A predicate object map must have at least one predicate map and one object map. Predicates must be valid IRIs, while objects may be IRIs, blank nodes, or literals. If the object is a literal value, a datatype or language tag may be defined. Object maps may be used to link triples maps through parent triples maps. Finally, predicate object maps may also be associated to named graphs.

# MAPPING QUALITY ASSESSMENT FRAMEWORK

This section presents our proposed framework for assessing and refining the quality of mappings used to generate and publish RDF datasets. The proposed framework is divided into two main stages: assessment and refinement and is presented in **Figure 1**.

The framework starts by analyzing a mapping file, which contains the rules on the transformation process for converting non-RDF data to RDF. The mapping is analyzed for quality issues through quality metrics. Each metric in our framework contains a declaratively description together with an implementation. This information is used in the generation of reports which detail the identified quality violations. The assessment report is used in the refinement process in order to adjust the mapping definitions. At this point, mapping engineers may select the appropriate refinement to each of the identified inconsistencies. It is also possible to finalize the framework or run the assessment process on the refined mappings. Finally, the framework generates a refined mapping containing the adjusted mappings and a report containing the identified violations. The proposed framework can be summarized as follows:

1. The mapping is assessed through the use of metrics for quality issues.
2. The framework executes each quality metric on the mapping and uses its description to generate information relating to violations identified within the mapping file into an assessment report.
3. The assessment report is then used to refine the mapping file. For each inconsistency, the framework allows mapping engineers to select the most appropriate refinement. The refinement process may be manual or semi-automatic and is described in detail in **Section 3.2.** At the end of this step, users may decide to execute the mapping assessment process on the refined mappings.
4. The process ends when there are no inconsistencies found in the mapping file or when users decide to finalize it. At this stage, the framework generates the final refined mapping and a mapping validation report.



**Figure 1:** Framework design.

We note that for some use cases, mapping engineers may want to select which metrics are required. It is also noted that users may want to finalize the process without adjusting all the identified inconsistencies. For these cases, the mapping validation report will capture such information.

## MAPPING QUALITY ASSESSMENT

This section describes the mapping quality assessment process within our proposed framework. The assessment process starts by loading the quality metrics contained within a library. This process also fetches the vocabularies defined within the mapping input into the framework. The information contained in those vocabularies can be used by quality metrics to identify schema related violations, as well as quality information related to the vocabulary itself, which we argue to be helpful to users defining mappings. For instance, a vocabulary may not include range or domain definitions for certain properties. Furthermore, this information is exposed by the framework to allow for the custom implemented metrics which require such information. Afterwards, the mapping is assessed using these metrics, thus generating an assessment report containing the identified violations within the mapping. The quality assessment within the framework design is split into three main processes, which are summarized as follows:

**Quality metrics.** The library containing the metrics being used during quality assessment. A number of these metrics have been derived from previous work, such as [29], which focus on metrics typically used for the quality of the resulting dataset. Others are proposed in this work for assessing specific mapping quality issues. These metrics are described in **Section 4**. The library of metrics can also be extended by users, allowing for customized metrics to be implemented.

**Assessment**. The quality metrics are loaded and the vocabularies used within the mapping definitions are fetched. As mentioned, the vocabularies are also exposed by the framework to third party implemented metrics. For instance, the information contained in vocabularies can be used to detect whether a mapping will produce a dataset with *domain and range* inconsistencies.

**Assessment report.** A report will be generated using the information contained in the loaded quality metrics (see **Section 4**)**.** This report describes violations detected by the quality metrics.

## REFINEMENT

This step is concerned with refining the identified quality issues within the mapping prior to generation of the RDF datasets. This process prevents violations within the mappings to be propagated to the resulting dataset [15]. Refinement can be executed in one of two ways: manual or semi-automatic refinement. The refinement method will depend on the violation being refined.

**Manual**. Manual refinements relate to violations, which the framework can identify, however, the framework is either unable to provide specific guidance on how the refinement is to be completed or the mapping engineer decides to adjust the identified inconsistency manually. As an example, a class definition used within the mapping does not have a human readable label or comment, which is considered a bad design practice [24] and thus a quality issue. The method which will be used to refine this violation is decided by the mapping engineer, which may be, for instance, the replacement of the class causing a violation to a different one. In this case, the mapping engineer manually provides which class to be used. Another possible adjustment would be for the vocabulary maintainers to add human readable information for the class being used. This type of adjustment must be executed in the vocabulary and thus is considered out of scope within the proposed mapping quality framework.

**Semi-automatic**. In this type of refinement, the framework provides a list of suggested adjustments, which may be used to refine the identified violation. For instance, if a mapping includes a class definition for foaf:person (with incorrect capitalization), however, this definition is not found in the vocabulary. The framework could search for similarly spelled class definitions within the vocabulary and recommend them to the mapping engineer. In this case, the framework retrieves the class foaf:Person and suggests it as an adjustment, which may or not be accepted by the mapping engineer, to replace the class in the mapping.

The refinement process ends when there are no inconsistencies found in the mapping or when users decide to finalize it. In the end, the final refined mapping and the mapping validation report are generated by the framework.

## OUTPUT

In the last step of the framework a mapping validation report and the refined mapping are generated. Violations within the mapping input into the framework may have been addressed by the refinement phase. The validation report contains information about the violations that were detected and adjusted within the original mapping. The generated artefacts are described as follows:

**Mapping validation report.** A machine-readable report is generated after the mapping file has been assessed and refined. This report contains information relating to the violations detected and adjusted by the framework. This report will give an indication of the overall quality of the mapping.

**Refined mapping.** The refined mapping is generated after the mapping engineer has refined the original mapping. As mentioned, this may be done either manually or semi-automatically, thus removing the identified inconsistencies contained in the mapping.

# QUALITY ASSESSMENT METRICS

This section presents metrics which are used within our framework for assessing quality in relation to the uplift mapping process, concerned with converting non-RDF resources to RDF.

The proposed mapping quality metrics draw inspiration from a survey covering various aspects of Linked Data quality [29]. In this survey, Linked Data quality assessment was defined as a methodology focused on evaluating whether a particular piece of data is relevant to a certain consumer. In this context, data quality problems refer to issues affecting the full potential of applications using that data [29]. The assessment of Linked Data quality was further classified into **categories, dimensions** and **metrics**. Categories are used to group data quality dimensions which are themselves made up of data quality metrics. Dimensions are abstract concepts used to define the "characteristics of a dataset" [29]. The dimensions used for our quality assessment framework, are described in **Table 1.** A data quality metric is thus defined as a procedure for measuring a data quality dimension.

The categories used in our framework are adopted from [29]. Therefore, it is noted that a category captures the same essence for the underlying dimensions that belong to that category. Nonetheless, these groups are not strictly disjoint and may be considered different depending on the applied use case scenarios. The four categories used in our framework are described as follows:

**Intrinsic category.** Intrinsic category contains the dimensions that are independent of the user’s context [29]. In the mapping process context, these dimensions focus on whether the mapping is correct (syntactically and semantically), whether the mapping is consistent in itself, and how complete it represents the data being mapped and the vocabularies being used.

**Representational category.** Representational category is concerned with the design of the data. In other words, metrics in this category evaluate how well the data is represented in terms of best practices and guidelines [9]. The dimensions in this category are mostly focused on the quality of the resulting datasets produced by mappings.

**Contextual category.** The contextual category groups dimensions and metrics highly depend on the context of the task at hand [9]. The dimensions for this category deal with trustworthiness and understandability. We note that the dimensions and metrics within this category do not specify how (i.e. which vocabularies, etc.) mappings and resulting datasets must define trustworthiness and understandability information. Nonetheless, existing work has shown how such information can be incorporated into mappings, which is argued to improve trustworthiness and understandability of mappings and datasets [11]. Thus, this category provides quality metrics related to detect whether such information is present.

**Accessibility category.** The accessibility category groups dimensions and metrics related to the access, authenticity and retrieval of data [14]. The dimensions for this category deal with licensing and availability and are derived from previous work on

capturing mapping provenance and metadata information [17].

Table 1: Categories and dimensions.

|  |  |  |
| --- | --- | --- |
| **Category** | **Dimension** | **Description** |
| Intrinsic | Data  Consistency (DC) | This dimension refers to the extent to which a dataset will be generated with no conflicting information. |
| Mapping consistency (MC) | This dimension refers to the extent to which a mapping is conformant to its mapping language. |
| RDF term correctness (RT) | This dimension refers to the extent to which a mapping correctly defines RDF terms. |
| Representational | Interpretability (IR) | This dimension is concerned to information being represented in an appropriate notation, and whether it is machine-processable. |
| Representational conciseness (RC) | This dimension refers to the representational of the resulting dataset being compact, well-formatted, and clear. |
| Contextual | Trustworthiness (TR) | This dimension refers to the extent to which data producers involved in the mapping process believe that the information in those mappings is "true". |
| Understandability (UT) | This dimension is concerned with human-readable information being provided to mappings and the resources being generated such that data producers and consumers are able to understand them. |
| Accessibility | Availability (AV) | This dimension refers to the extent to which the mapping, the mapped data, and the resulting dataset are available. |
| Licensing (LI) | This dimension refers to the license under which a mapping and its resulting dataset can be (re) used. |

The metrics described in the following sections, are grouped into different dimensions. Each of the metrics described in **Table 2**, **Table 3** and **Table 4**, have been annotated, below their ID, using the dimension ID’s shown in **Table 1**.

We furthermore split our mapping quality assessment into three different aspects of quality, which are mapping, data and vocabulary quality. Finally, we note that a number of the metrics defined in this section are focused on R2RML, which is a W3C Recommendation for transforming relational databases to RDF (for more information about R2RML we refer the reader to the W3C specification [7]). We note that the proposed metrics only assess the mappings and not the source data which means violations may still be present within the resulting dataset.

## MAPPING QUALITY ASPECT

Firstly, we consider the quality of the mapping itself, by assessing, for instance, the extent to which the mapping correctly conforms to the specification, of the mapping language used and also whether there are no redundancy definitions within the mapping (which would affect the performance of mapping engines). The metrics relating to this aspect, were inspired by [6,7,14,21] and are described in **Table 2**. Violations identified using these metrics may be refined manually or semi-automatically. For instance, the framework cannot add the missing information, such as a *missing predicate*. The framework does, however, provide information about the identified violation, and may even be able to suggest refinements based on the vocabularies being used. Another example is metric MP7, as R2RML defines only three term types (rr:IRI, rr:Literal, rr:BlankNode) for which one may be suggested as a refinement to the mapping engineer.

**Table 2:** Mapping quality metrics.

|  |  |  |
| --- | --- | --- |
| **ID** | **Metric** | **Description** |
| MP1  (MC) | Valid language datatype definition | Objects maps with literal values may refer to only one datatype and no language tags [6,7,15]. |
| MP2  (MC) | Valid language tag definition | Object maps with literal values may refer to only one language tag and no datatype [6,7,15]. |
| MP3  (MC) | Valid subject map definition | One subject map, which may have zero or more class definitions [6,7,15,22]. |
| MP4  (MC) | Valid predicate object map definition | There must exist at least one predicate map and one object map. These are used to generate the predicates and objects of the triples [6,7,15,22]. |
| MP5  (MC) | Valid parent triples map definition | The triples map being referenced must exist in the mapping and, when defined, join conditions must have both parent and child column definitions [6,7,15]. |
| MP6  (MC) | Valid logical table definition | A logical table exist and references either a table (or view) or an SQL query [6,7,15]. |
| MP7  (MC) | Valid term type definition | Terms maps are assigned the correct term types. Subject maps may be IRIs or blank nodes, predicate maps must be IRIs, and object maps may be IRIs, blank nodes, or literal [6,7,15]. |
| MP8  (RT) | Valid subject definition | Subject definitions must be valid URIs unless its type is defined as blank node [6,7,15]. |
| MP9  (RT) | Valid predicate definition | The predicate definition is a valid URI [6,7,15]. |
| MP10  (RT) | Valid named graph definition | The named graph definition is a valid URI [6,7,15]. |
| MP11  (RT) | Valid datatype definition | The datatype definition is a valid URI. In R2RML, this would involve validating object maps associated with datatypes [6,7,15]. |
| MP12  (RT) | Valid literal language tags | Valid language tags are defined as per RFC 5646 (BCP 47)[[1]](#footnote-2) [6,7,15]. |
| MP13  (RC) | Duplicate triples defined | Mappings which generate the same triple more than once [4]. |

## DATA QUALITY ASPECT

The second aspect relates to the quality of the output generated by an engine processing the input data and the mapping. Poor design decisions made at the stage of defining a mapping, such as using *non-dereferenceable[[2]](#footnote-3) classes and properties*, or *deprecated* ones, will decrease the quality of the dataset. This aspect focuses on the quality of the output data which can be identified and fixed during mapping design-time. In addition, assessing mapping quality from a design perspective would avoid the propagation of violations to all datasets generated by a particular mapping. The metrics relating to this aspect were inspired and adopted from [9,16,28–30] and are described in **Table 3**. Our framework can semi-automatically refine metrics (D4, D6, D7), however, the other data quality metrics must be refined manually.

Table 3: Data quality metrics.

|  |  |  |
| --- | --- | --- |
| ID | Metric | Description |
| D1  (RC) | Minimal usage of RDF structures | The usage of RDF structures is discouraged due to their complexity [9,16,29,30]. |
| D2  (RC) | No query parameters in URI's | The use of query parameters is not recommended by the W3C best practices for URIs [9,28–30]. |
| D3  (IR) | Usage of blank nodes | Usage of blank nodes is discouraged as they cannot be externally referenced [9,29,30]. |
| D4  (IR) | Usage of undefined classes and properties metric | Classes and properties are considered undefined when it is not possible to dereference them against their namespace [9,29,30]. |
| D5  (DC) | No use of entities as members of disjoint classes | Individuals of one class cannot be simultaneously members of another class [9,29,30]. |
| D6  (DC) | Usage of Incorrect Domain or Range Types | This will validate that the correct domain and range are being used in the mapping definition  [9,29,30]. |
| D7  (DC) | Usage of Incorrect Data type and object properties | This will validate that all objects have been assigned the correct datatype [9,29,30]. |

## 

## VOCABULARY QUALITY ASPECT

Finally, the third aspect we consider relates to the quality of the vocabularies used within the mapping, by assessing for instance, that these classes and properties contain human readable labels or comments in the vocabulary. The rationale for including this aspect is to ensure the quality of the resulting datasets by making quality information related to the vocabularies being used in the mapping transparent to mapping engineers. The metrics relating to this aspect, were inspired by [3,9,20,23,25,29,30] and are described in **Table 4**. Mapping engineers, however, may decide to use different classes and properties from other vocabularies based on the quality violations which are made explicit through these quality metrics. Furthermore, we argue that these metrics are important as they make quality information related to the vocabularies explicit to mapping engineers and to consumers of the resulting datasets.

**Table 4:** Vocabulary quality metrics.

|  |  |  |
| --- | --- | --- |
| ID | Metric | Description |
| VOC1  (UT) | Human readable labels/comments | Humans consuming the information should be able to understand the Linked Data Resource [9,20,23,25,29,30]. |
| VOC2  (UT) | Domain and range definitions. | A property should have a range and domain definition [21,22,26,27]. |
| VOC3  (TR) | Basic Provenance Information | Consumers need to understand where the data has originated [3,9,20,29,30]. |
| VOC4  (LI) | Machine readable licensing | Allows the licensing information to be queried by machines [9,20,23,29,30]. |
| VOC5  (LI) | Human readable licensing | Allows humans to read and understand license in a textual format [9,20,23,29,30]. |
| VOC6  (AV) | Dereferencability of the URI | Ensuring the resources are retrievable is one of the main principles of Linked Data [3,9,20,29,30]. |
| VOC7  (UT) | Regular Expression Definition of a URI | Allows agents to interpret and explore resources better [9,29,30]. |
| VOC8  (UT) | Indication of Used Vocabularies | Indicating the vocabularies used in the datasets allows humans to query them [9,23,29,30]. |

We note that our work may consider a mapping to be high quality, however, the resulting data may still contain inconsistencies as the mapping is only assessed in our framework and not the source data. It is not intended that our framework replaces existing quality assessment frameworks, but instead it extends quality assessment to cover the data generation process.

# FRAMEWORK IMPLEMENTATION

Our implementation of the proposed quality assessment framework is specifically designed to assess and refine R2RML mappings. We have chosen to target the R2RML mapping language, for being the W3C-Recommended language for transforming relational databases to RDF.

Our quality assessment framework implementation[[3]](#footnote-4) was designed using a combination of core SHACL [17], SHACL advanced features[[4]](#footnote-5), JavaScript, Java and SPARQL to support the refinement of mappings. SHACL allows for the semantic description of constraints, which are used to describe quality issues in our framework. SHACL advance features allows users to define custom constraints, which are required for the definition of a number of quality metrics defined in our framework. Additionally, SHACL produces validation reports containing information on the identified issues which can be queried and processed during the refinement of those violations. For our implementation, we are using SHACL 1.3.0[[5]](#footnote-6) and Java 11.0.6.

## IMPLEMENTATION

A diagram of our implementation is shown in **Figure 2.** The steps involved in our implementation are summarized as follows:

**R2RML mapping**. The file containing the R2RML mapping definition. This mapping will be assessed and refined by our framework.

**Quality metrics.** This library contains the quality metrics, which are described and implemented using SHACL [17] core and SHACL advanced features (JavaScript extension[[6]](#footnote-7)). The metrics implemented are drawn from those metrics described in **Section 4**.

**JavaScript Engine.** A JavaScript engine is required for running the JavaScript API used by the SHACL JavaScript extension. SHACL uses the Nashorn JavaScript engine[[7]](#footnote-8), for running the JavaScript API. However, since, this engine will not have access to functions provided by browsers, which we will require, for instance to dereference URIs, we will invoke a number of Java methods within our Nashorn environment.

**SPARQL Refinemen**t Since R2RML is described using RDF, we use SPARQL[[8]](#footnote-9) to refine the mappings. SPARQL allows us to retrieve and manipulate information from vocabularies and mappings.

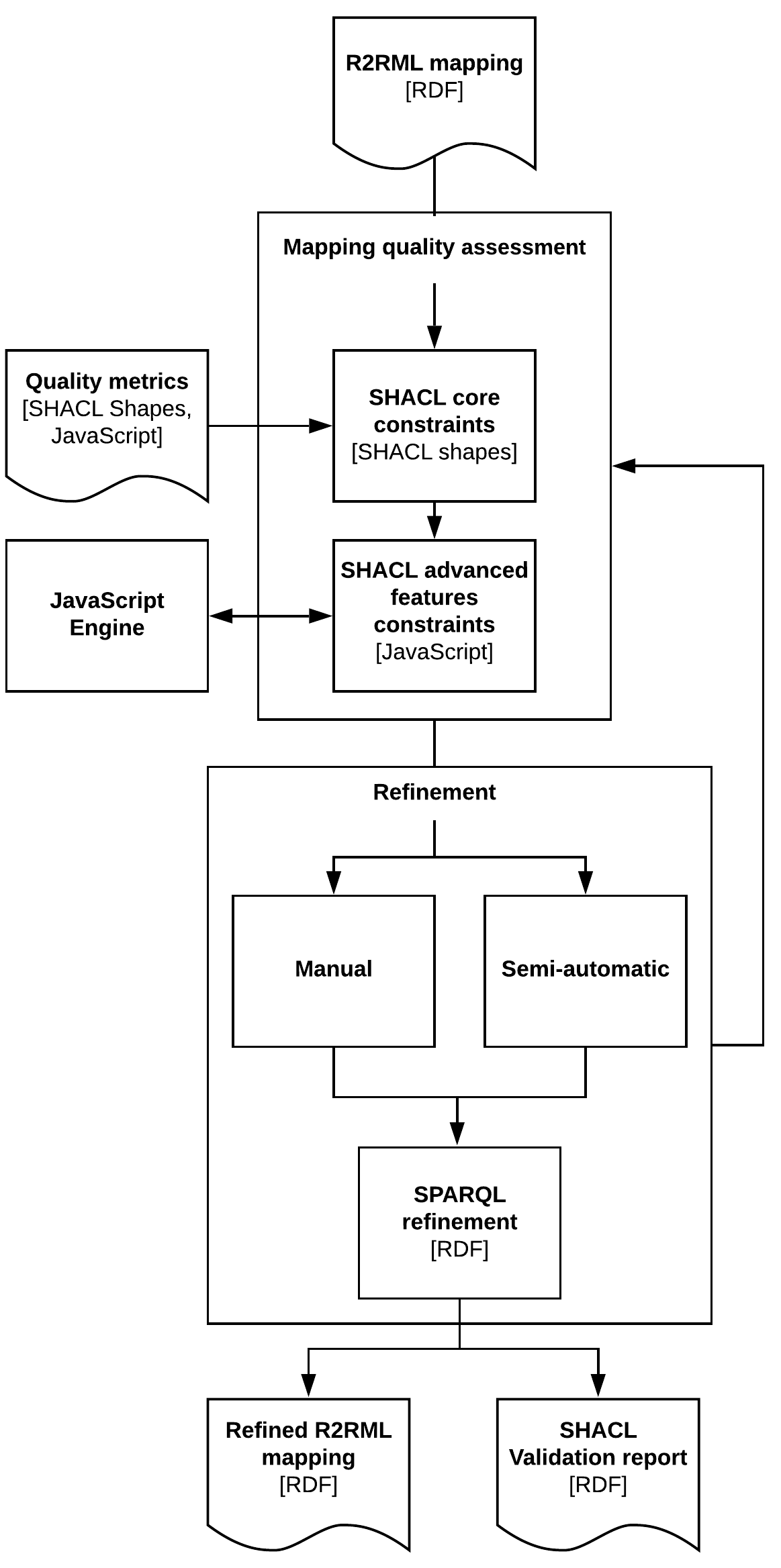


Figure 2: Framework implementation diagram.

## DEMONSTRATION WALK THROUGH

To demonstrate our framework and implementation, we have a created a sample mapping, which is shown in **Listing 1**. This will be used as our running example during our demonstration of our framework in the following sections. This simple illustrative example is deliberately designed to show different metrics and features of the assessment framework in action. The mapping we have created is based around a scenario, where we take data about people, which is contained within a personnel database and uplift this data to RDF using R2RML. The mapping used during this process will be assessed and refined in the following sections. This will ensure violations are identified and addressed prior to generating the RDF dataset.

## R2RML MAPPING

In the R2RML mapping shown in **Listing 1**, which defines the source table as ***“people”****.* The mapping also defines the subjects of the triples to be instances of the class dbo:Agent. The predicate dbo:ageis related to values from the column ***“age”*** and have the datatype xsd:date. A second predicate, ex:name, has an IRI term type (rr:IRI) is related to values coming from the attribute column ***“name”.***

|  |
| --- |
| <#TripleMap1>  rr:logicalTable [ rr:tableName “people”];  rr:subjectMap [  rr:template "http://example.org/person/{id}";  rr:class dbo:Agent;  ];  rr:predicateObjectMap [  rr:predicate dbo:age;  rr:objectMap [  rr:column"age";  rr:datatype xsd:date; ];  ];  rr:predicateObjectMap [  rr:predicate ex:name;  rr:termType rr:Literal;  rr:objectMap [ rr:column "name"; ];  ];. |

Listing 1: R2RML mapping definition.

## MAPPING QUALITY ASSESSMENT

The framework starts by assessing the mapping using the quality metrics outlined in **Section 4**. After assessing the mapping, the framework generates a machine-readable validation report containing the identified quality violations. We will discuss the three following violations detected by our framework when assessing the mapping shown in **Listing 1**.

1. **Incorrect term type.** The term type for a predicate object map should be an IRI (rr:IRI), however, it has been defined as a literal (rr:Literal).
2. **Dereferencability of the URI.** The resource ex:namecannot be retrieved.
3. **Incorrect datatype.** The datatype for the object of a triple with predicate dbo:ageis xsd:integer, instead of the datatype defined in the mapping (xsd:date).

An excerpt of the validation report generated from our framework by assessing the mapping shown in **Listing** **1** is shown in **Listing 2.**

|  |
| --- |
| [ sh:result [  a sh:ValidationResult ;  sh:focusNode \_:b0 ;  sh:resultMessage "Incorrect datatype definition detected." ;  sh:resultSeverity sh:Violation ;  sh:sourceConstraint \_:b1 ;  sh:sourceConstraintComponent sh:JSConstraintComponent ;  sh:sourceShape ex:ValidDatatypeShape ;  sh:value dbo:age ] ] . |

**Listing 2:** Excerpt of the generated validation report.

## METRIC IMPLEMENTATION

In the following subsections we will discuss how our framework implements the metrics that have identified those violations from the mapping presented in **Listing 1**.

### Incorrect term type

Verifying the term type for a predicate map is can be executed using purely SHACL core constraints. An excerpt of the SHACL code used to identify this inconsistency is shown in **Listing 3**.

|  |
| --- |
| ex:PredicateShape a sh:NodeShape ;  sh:targetObjectsOf rr:predicateObjectMap ;  sh:property [  sh:path rr:termType ;  sh:in ( rr:IRI ) ;  ] ;. |

Listing 3: Excerpt of valid term type metric.

The SHACL code shown in **Listing 3** uses the SHACL in clause (sh:in) to ensure the term type defined within a predicate map is an IRI (rr:IRI).

### Dereferencability of the URI

Verifying whether a URI resource, which is defined within the mapping, is dereferencable, requires retrieving the resource and validating that the resource contains its description. The steps involved in verifying that a URI resource defined with a mapping is dereferencability is summarized as follows:

1. A HTTP request is carried out, using the URI, which requests the RDF format of the resource.
2. The resource retrieved is verified, to ensure it contains an RDF resource.

### Incorrect Datatype

Comparing the datatype assigned to an object with a given predicate requires retrieving the vocabulary for the given predicate. The steps involved in verifying the correct datatype is defined within the mapping definition are summarized as follows:

1. SHACL executes a call to the SHACL JavaScript API.
2. The JavaScript API calls the custom *validateDatatype* function, however, in this case the JavaScript engine being used does not have access to functions provided by browsers, since we need to fetch the vocabulary, a Java method is required.
3. A Java method is called by the JavaScript API with the dbo:age predicate. This method then fetches the predicates vocabulary. Afterwards, a SPARQL ask query is executed to verify that the xsd:datedatatype, which is assigned to the object of the triple with the dbo:agepredicate within the mapping,matches the datatype within the vocabulary, which is xsd:integer*.*

## REFINEMENT

Quality issues within the mapping should be refined when possible, thus removing quality issues and producing a higher quality mapping. In this section we will discuss the refinement capabilities of our framework relating to the violations identified in the previous section. Refining the incorrect term type contained within in the mapping can be completed either manually or semi-automatically, as described in **Section 5.6.1**. Refining the dereferencability of the URI, can be refined manually by the mapping engineer or the vocabulary maintainers, as described in **Section 5.6.2**. Refining the incorrect datatype definition, contained within the mapping, can be completed either manually or semi-automatically, as described in **Section 5.6.3**.

### Incorrect term type

Refining an incorrect term type for a predicate map can be undertaken semi-automatically as the R2RML specification [7] states that the term type for a predicate map must be an IRI (rr:IRI). The framework will suggest a modification which changes this term type to an IRI (rr:IRI) and provide a SPARQL delete/insert operation[[9]](#footnote-12) which could be executed on this mapping.

### Dereferencability of the URI

If a URI contained within the mapping is not dereferencable, the mapping engineer can refine this violation manually, or this violation could also be fixed by the vocabulary maintainers. For the former, a mapping engineer could remove the ex:name property which is contained within the mapping and insert a property which can be dereferenced. In the latter, the vocabulary maintainers could resolve any dereferenceability issues.

### Incorrect Datatype

Refining an incorrect datatype may be undertaken semi- automatically. The framework will suggest to the mapping engineer an adjustment that could refine this violation. Since the datatype for dbo:age is defined as xsd:integer within its vocabulary, the framework can suggest changing the datatype definition within the mapping to xsd:integer. If the mapping engineer accepts the suggested modification, the framework will construct and execute the SPARQL delete/insert operation shown in **Listing 5** after it has retrieved the correct datatype, using the SPARQL query shown in **Listing 4.**

|  |
| --- |
| SELECT ?datatype  WHERE { dbo:age rdfs:range ?datatype. } |

Listing 4: SPARQL select query to retrieve datatype.

|  |
| --- |
| DELETE{?om rr:datatype ?datatype}  INSERT{?om rr:datatype xsd:integer }  WHERE {  ?sub rr:predicateObjectMap ?pom.  ?pom rr:predicate dbo:age.  ?pom rr:objectMap ?om.  ?om rr:datatype ?datatype. } |

Listing 5: SPARQL delete/insert operation to change datatype.

# RELATED WORK

Several quality assessment frameworks have been proposed in literature to assess Linked Data datasets. Luzzu [8] is one such example, which is defined as an extendible, interoperable, scalable, and customisable quality assessment framework for Linked Data. RDF Unit [18] uses SPARQL query templates to provide a test-driven quality assessment. LINK-QA [12] is a framework which is focused on the quality of links between Linked Data datasets. LD Sniffer [21] is a quality assessment framework which focuses on the accessibility of Linked Data i.e. the deferenciability of resources. Other approaches have focused on the quality of ontologies and vocabularies. For example, OOPS! (Ontology Pitfall Scanner!) [24] is a web based framework which validates ontologies. This framework uses a catalogue of common pitfalls and can be used outside of the ontology development platform. OntoQA [27] is another approach which uses different dimensions to evaluate ontology schemas. Finally, OntoMetrics [19] is a web-based system to validate and display statistics about ontologies.

In relation to mapping quality assessment, an approach has been proposed in [10]. In this work, the authors have extended RDFUnit [18] to also cover the validation of mappings against its vocabularies and ontologies. RDFUnit, however, relies on SPARQL queries and therefore cannot assess all mapping quality aspects. As an example, the assessment of undefined classes and properties cannot be computed by RDFUnit as SPARQL does not deference resources natively. In order to address these short comings, another approach extended Luzzu for the task of assessing the aspects of a mapping which would affect the quality of the final datasets [15]. Resglass [13] is a framework which uses rules to detect inconsistencies within mappings and the resulting dataset. Resglass, however, does not allow for customized quality metrics and refinements. Our proposed framework is concerned with both mapping quality as well as aspects within a mapping which affect the generated dataset, such as the quality of the vocabularies being used. Moreover, by implementing metrics using SHACL we provide users with both human and machine-readable information about the metrics and the identified quality issues which are then used to refine those mappings.

# CONCLUSION AND FUTURE WORK

Several quality assessment frameworks have been proposed in literature,but in most cases, these do not focus on RDF generation and publication. This paper tackles this issue by proposing a quality framework focused on the processes related to the definition of mappings which are used to generate and publish RDF datasets. The introduction of quality assessment and refinement procedures as part of the mapping process is expected to aid data producers into creating high quality mappings and datasets. This paper also provides an implementation of the proposed framework, where we have defined a number of quality metrics focused on different aspects which affect the quality of mappings. Finally, the proposed framework was demonstrated through a walk-through use case.

Future work includes the definition and implementation of other metrics in order to cover a greater number of quality issues. Future work will also include the implementation of more expressive refinement capabilities. We also plan on evaluating our framework by comparing it existing ones, such as Resglass. Finally, we also plan on extending our implementation to support mappings that transform data from other data formats, such as CSV and XML.

ACKNOWLEDGMENTS

This research was conducted with the financial support of the SFI AI Centre for Research Training under Grant Agreement No. 18/CRT/6223 at the ADAPT SFI Research Centre at Trinity College Dublin. The ADAPT SFI Centre for Digital Media Technology is funded by Science Foundation Ireland through the SFI Research Centres Programme and is co-funded under the European Regional Development Fund (ERDF) through Grant # 13/RC/2106.

REFERENCES

[1] Christian Bizer and Tom Heath. *Linked Data - The Story So Far*. Retrieved from http://linkeddata.org/docs/ijswis-special-issue

[2] Christian Bizer, Tom Heath, Kingsley Idehen, and Tim Berners-Lee. 2008. Linked data on the web (LDOW2008). *Proc. 17th Int. Conf. World Wide Web (WWW ’08)* (March 2008). DOI:https://doi.org/10.1145/1367497.1367760

[3] Andrew Burton-Jones, Veda C. Storey, Vijayan Sugumaran, and Punit Ahluwalia. 2005. A semiotic metrics suite for assessing the quality of ontologies. In *Data and Knowledge Engineering*, 84–102. DOI:https://doi.org/10.1016/j.datak.2004.11.010

[4] David Chaves-Fraga, Kemele M. Endris, Enrique Iglesias, Oscar Corcho, and Maria-Esther Vidal. 2019. What Are the Parameters that Affect the Construction of a Knowledge Graph? . 695–713. DOI:https://doi.org/10.1007/978-3-030-33246-4\_43

[5] Ademar Crotti, Christophe Debruyne, Rob Brennan, and Declan O’Sullivan. 2017. An evaluation of uplift mapping languages. *Int. J. Web Inf. Syst.* 13, 4 (2017), 405–424. DOI:https://doi.org/10.1108/IJWIS-04-2017-0036

[6] Ademar Crotti, Christophe Debruyne, and Declan O’Sullivan. 2018. Juma Uplift: Using a Block Metaphor for Representing Uplift Mappings. In *Proceedings - 12th IEEE International Conference on Semantic Computing, ICSC 2018*, Institute of Electrical and Electronics Engineers Inc., 211–218. DOI:https://doi.org/10.1109/ICSC.2018.00037

[7] Souripriya Das, Seema Sundara, and Richard Cyganiak. 2012. R2RML: RDB to RDF Mapping Language. *W3C Recomm.* September 2012 (2012), 1–34. DOI:https://doi.org/10.1017/CBO9781107415324.004

[8] Jeremy Debattista, Christoph Lange, and Sören Auer. 2014. Luzzu - A Framework for Linked Data Quality Assessment. (December 2014). Retrieved from http://arxiv.org/abs/1412.3750

[9] Jeremy Debattista, Christoph Lange, Sören Auer, and Dominic Cortis. 2018. Evaluating the quality of the LOD cloud: An empirical investigation. *Semant. Web* 9, (March 2018), 1–43. DOI:https://doi.org/10.3233/SW-180306

[10] Anastasia Dimou, Dimitris Kontokostas, Markus Freudenberg, Ruben Verborgh, Jens Lehmann, Erik Mannens, Sebastian Hellmann, and Rik Van de Walle. 2015. Assessing and refining mappings to RDF to improve dataset quality. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, 133–149. DOI:https://doi.org/10.1007/978-3-319-25010-6\_8

[11] Anastasia Dimou, Tom De Nies, Ruben Verborgh, Erik Mannens, and Rik de Walle. 2016. Automated Metadata Generation for Linked Data Generation and Publishing Workflows. In *Proceedings of the 9th Workshop on Linked Data on the Web* (CEUR Workshop Proceedings).

[12] Christophe Guéret, Paul Groth, Claus Stadler, and Jens Lehmann. 2012. Assessing linked data mappings using network measures. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 87–102. DOI:https://doi.org/10.1007/978-3-642-30284-8\_13

[13] Pieter Heyvaert, Ben De Meester, Anastasia Dimou, and Ruben Verborgh. 2019. Rule-driven inconsistency resolution for knowledge graph generation rules. *Semant. Web* 10, 6 (2019). DOI:https://doi.org/10.3233/SW-190358

[14] Aidan Hogan, Jürgen Umbrich, Andreas Harth, Richard Cyganiak, Axel Polleres, and Stefan Decker. 2012. An empirical survey of Linked Data conformance. *J. Web Semant.* 14, (2012), 14–44. DOI:https://doi.org/10.1016/j.websem.2012.02.001

[15] Ademar Crotti Junior, Jeremy Debattista, and Declan O’Sullivan. 2019. Assessing the Quality of R2RML Mappings. In *Joint Proceedings of the 1st International Workshop On Semantics ForTransport and the 1st International Workshop on Approaches for MakingData Interoperable co-located with 15th Semantics Conference (SEMANTiCS2019), Karlsruhe, Germany, September 9, 2019* (CEUR Workshop Proceedings), CEUR-WS.org. Retrieved from http://ceur-ws.org/Vol-2447/paper2.pdf

[16] Pavel Klinov and Dmitry Mouromtsev. 2015. Measuring the Quality of Relational-to-RDFMappings. In *Communications in Computer and Information Science*, Springer Verlag. DOI:https://doi.org/10.1007/978-3-319-24543-0

[17] Holger Knublauch and Dimitris Kontokostas. 2017. Shapes Constraint Language (SHACL), W3C Recommendation 20 July 2017. *URL https//www. w3. org/TR/shacl* (2017).

[18] Dimitris Kontokostas, Patrick Westphal, Sören Auer, Sebastian Hellmann, Jens Lehmann, Roland Cornelissen, and Amrapali Zaveri. 2014. Test-driven evaluation of Linked Data quality. In *WWW 2014 - Proceedings of the 23rd International Conference on World Wide Web*, Association for Computing Machinery, Inc, 747–757. DOI:https://doi.org/10.1145/2566486.2568002

[19] Birger Lantow. 2016. OntoMetrics: Putting Metrics into Use for Ontology Evaluation. 186–191. DOI:https://doi.org/10.5220/0006084601860191

[20] Bernadette Farias Lóscio, C Burle, and N Calegari. 2016. Data on the web best practices. W3C recommendation. *World Wide Web Consort.* (2016).

[21] Nandana Mihindukulasooriya, Raúl García-Castro, and Asunción Gómez-Pérez. 2017. LD sniffer: A quality assessment tool for measuring the accessibility of linked data. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, 149–152. DOI:https://doi.org/10.1007/978-3-319-58694-6\_20

[22] Christoph Pinkel, Carsten Binnig, Peter Haase, Clemens Martin, Kunal Sengupta, and Johannes Trame. 2014. How to best find a partner? An evaluation of editing approaches to construct R2RML mappings. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, 675–690. DOI:https://doi.org/10.1007/978-3-319-07443-6\_45

[23] Mar\’{\i}a Poveda-Villalón, Bernard Vatant, Mari Carmen Suárez-Figueroa, and Asunción Gómez-Pérez. 2013. Detecting Good Practices and Pitfalls When Publishing Vocabularies on the Web. In *Proceedings of the 4th International Conference on Ontology and Semantic Web Patterns - Volume 1188* (WOP’13), CEUR-WS.org, Aachen, DEU, 39–51. Retrieved from data quality metrics ontology testing

[24] María Poveda-Villalón, Asunción Gómez-Pérez, and Mari Carmen Suárez-Figueroa. 2014. OOPS! (OntOlogy Pitfall Scanner!): An On-line Tool for Ontology Evaluation. *Int. J. Semant. Web Inf. Syst.* 10, 2 (2014), 7–34.

[25] María Poveda-Villalón, Mari Carmen Suárez-Figueroa, and Asunción Gómez-Pérez. 2012. Validating ontologies with OOPS! In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 267–281. DOI:https://doi.org/10.1007/978-3-642-33876-2\_24

[26] Gezim Sejdiu, Anisa Rula, Jens Lehmann, and Hajira Jabeen. 2019. A Scalable Framework for Quality Assessment of RDF Datasets. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer, 261–276. DOI:https://doi.org/10.1007/978-3-030-30796-7\_17

[27] Samir Tartir, I. Budak Arpinar, Michael Moore, Amit P. Sheth, and Boanerges Aleman-meza. 2005. OntoQA: Metric-based ontology quality analysis. In *IEEE ICDM 2005 Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources*. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.131.2087

[28] Olivier Théreaux. 2003. Common http implementation problems. W3C note. *World Wide Web Consortium. http//www. w3. org/TR/chips* (2003), 49.

[29] Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, Jens Lehmann, and Sören Auer. 2015. Quality assessment for Linked Data: A Survey. *Semant. Web* 7, (March 2015), 63–93. DOI:https://doi.org/10.3233/SW-150175

[30] Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, Jens Lehmann, Sören Auer, and Pascal Hitzler. 2013. Quality assessment methodologies for linked open data. *Submitt. to Semant. Web J.* 1, (2013), 1–5. DOI:https://doi.org/10.3233/SW-150175

1. http://www.rfc-editor.org/info/rfc5646 [↑](#footnote-ref-2)
2. Linked Data principles recommend that all URIs in a dataset are deferenceable i.e. that HTTP clients are able of accessing and retrieving the resources identified by such URIs. [↑](#footnote-ref-3)
3. https://github.com/alex-randles/Mapping-quality-assessment-framework [↑](#footnote-ref-4)
4. https://www.w3.org/TR/shacl-af/ [↑](#footnote-ref-5)
5. https://github.com/TopQuadrant/shacl/releases/tag/shacl-1.3.0 [↑](#footnote-ref-6)
6. https://www.w3.org/TR/shacl-js/ [↑](#footnote-ref-7)
7. https://en.wikipedia.org/wiki/Nashorn\_(JavaScript\_engine) [↑](#footnote-ref-8)
8. https://www.w3.org/TR/sparql11-query/ [↑](#footnote-ref-9)
9. https://github.com/alex-randles/Mapping-quality-assessment-framework/blob/master/resources/example\_refinement.rq [↑](#footnote-ref-12)