Harmonizing the OQuaRE Quality Framework

Achim Reiz[a] and Kurt Sandkuhl[b]
Chair of Information Systems, Rostock University, Albert-Einstein-Str. 22, 18059 Rostock Germany

Keywords: Ontology, Quality, Quality Framework, OQuaRE, Semantic

Abstract: Measuring ontology quality using metrics is far from a trivial task – one has to pick the right metrics for the right task and then interpret these values in a meaningful way. Without help, these interpretations are often highly subjective, even for trained knowledge engineers. Quality frameworks can assist and objectify the evaluation. One of the more prominent frameworks in ontology evaluation is OQuaRE, which builds upon the SQuaRE standard for software evaluation. Not only provides it tangible metrics for assessing an ontology, but it also suggests an interpretation for these values in the form of a quality rating and links these metrics to a broader quality framework.
However, during an implementation effort, the authors identified some drawbacks. In the last years, various metrics have been proposed that sometimes seem to conflict with each other or are inconclusive in their descriptions. The resources on the quality framework are distributed over web pages and papers. The following paper aims first to present the drawbacks the framework currently has. At the next step, we resolve the current heterogeneities and collect the information of the various sources. We aim to provide a one-stop information resource on OQuaRE to enable our further research and applications efforts.

1 INTRODUCTION

The selection, building, and integration of ontologies are far from trivial tasks. The idea of building shared knowledge bases emphasizes the reusing of already accepted terminologies. As vast quantities of ontologies have been developed over time, how does one find the right ontology with the best quality for the individual use case? Moreover, how can we help the developer create high-quality artifacts during the ontology development process? Automated quality metrics can assist the knowledge engineer in the selection and development process. It enables to grasp differences between two ontologies or two ontology versions.

There are a lot of different quantifiable attributes in an ontology that one can use, like attributes that are concerned with properties of the graph, the amount of human-centered annotations, the diversity of relations, and much more. Quality frameworks provide orchestration and meaning to these otherwise arbitrary and isolated measurement points. Over the past years, some ontology quality frameworks have been proposed:

Tartir et al. developed OntoQA, proposing schema, instance, relationship, and class-specific metrics (Tartir et al., 2005). Gangemi et al. proposed a large variety of primarily graph-related measurements (Gangemi et al., 2005), and Yao et al. (Yao et al., 2005) presented a set of metrics to measure cohesion. Furthermore, OQuaRE, initially proposed by Duque-Ramos et al., developed a quality framework based on the SQuaRE software methodology (Duque-Ramos et al., 2011).

OQuaRE was first introduced in 2011. Since then, it has been used by several publications involving, among others, always Duque-Ramos and Fernandez-Breis (Duque-Ramos et al., 2011; Duque-Ramos et al., 2013; Duque-Ramos et al., 2014; Duque-Ramos et al., 2016; Franco et al., 2020; M. Quesada-Martínez et al., 2015; Manuel Quesada-Martínez et al., 2017).

OQuaRE is probably the most holistic framework compared to the proposals by the authors named above. The proposed metrics are mapped to a rating system, showing which values ranges are desirable. Further, the metrics are associated with quality characteristics

[a] https://orcid.org/0000-0003-1446-9670
[b] https://orcid.org/0000-0002-7431-8412

Reiz, A. and Sandkuhl, K.
Harmonizing the OQuaRE Quality Framework.
DOI: 10.5220/0011077200003179
In *Proceedings of the 24th International Conference on Enterprise Information Systems (ICEIS 2022) - Volume 2*, pages 148-158
ISBN: 978-989-758-569-2; ISSN: 2184-4992

like *reusability* or *portability*. The authors also provide an online calculation tool (Fernandez-Breis et al., 2018), online documentation (*OQuaRE: A SQuaRE Based Quality Evaluation Framework for Ontologies)*, and a wiki (*OQuaRE Wiki*, 2016).

Most other frameworks only propose the metrics without stating how it affects these quality characte/-ristics. None of the competing proposals provide such detailed interpretations for the given metrics in the form of a school-grade-like rating system. Further, metric implementations in software tools are scarce (Reiz et al., 2020). Thus, the knowledge engineer who needs an evaluation often has no means to calculate the proposed metrics. All these factors contribute to the relevance of OQuaRE as practical, applicable quality guidance for ontology engineering.

In an effort to collect and map the various metric frameworks, their similarities, and differences as part of a larger research project (Reiz, 2020), we started to model the proposed quality dimensions to later create a shared metric interface for the different frameworks. However, for the OQuaRE-framework, that proved to be challenging: Over time, an increasing amount of metrics have been proposed in various publications. Some of the metrics were altered over time, and others are vague in their definition. The full extent of the documentation is available on, at times contradicting, online resources (*OQuaRE Wiki*, 2016; *OQuaRE: A SQuaRE Based Quality Evaluation Framework for Ontologies)*. These discovered limitations made the planned application of the framework difficult.

This paper targets to collect, harmonize, and precise OQuaRE. We aim to build a solid foundation for our further use and investigation of the framework. This research further shall enable other researchers and knowledge engineers to implement the same version of the metrics and make future results comparable. At first, we present the heterogeneous metrics, precise and harmonize them. In the next step, we collected and compared the various sources and presented the framework to the full extent.

2 IDENTIFIED HETEROGENEITIES

The different papers referencing OQuaRE propose 19 ontology metrics, even though not all are referenced and used in every paper. For this section, we checked whether the quality metrics proposed in the papers (Duque-Ramos et al., 2011; Duque-Ramos et al., 2013; Duque-Ramos et al., 2014; Duque-Ramos et al., 2016; M. Quesada-Martínez et al., 2015; Manuel Quesada-Martínez et al., 2017), in the online documentation and wiki (*OQuaRE Wiki*, 2016; *OQuaRE: A SQuaRE Based Quality Evaluation Framework for Ontologies)*, and the tool (Fernandez-Breis et al., 2018) are consistent with each other.

Twelve of the OQuaRE metrics are well defined. However, six of the metrics were proposed differently by the newer papers, even though they sometimes recalled the previous ones as their foundation. One metric was described homogeneously, but its definition seems ambiguous.

To homogenize the papers, we selected the metrics with the most acceptance in the community measured by citations. This approach emphasizes the definitions by (Duque-Ramos et al., 2011), cited 67 times (at the time of writing this paper), then (Duque-Ramos et al., 2013) and its associated documentation (*OQuaRE: A SQuaRE Based Quality Evaluation Framework for Ontologies)* with 34 citations, and (Duque-Ramos et al., 2014) with 15 citations.

2.1 NOCOnto (Number of Children)

The first metric that seems inconsistent in its definitions is NOCOnto. It is defined by (Duque-Ramos et al., 2016; M. Quesada-Martínez et al., 2015) as the *"Mean number of direct **subclasses** per class minus the **subclasses** of thing"*. (Duque-Ramos et al., 2011), as well as the documentation web page (*OQuaRE: A SQuaRE Based Quality Evaluation Framework for Ontologies)* proposes the same metric but uses the name "relationship" for the direct sub-class *relations*: *"Mean number of direct subclasses. It is the number of **relationships** divided by the number of classes minus the **relationships** of Thing"*. However, as this paper consistently uses the word "*relationship*" where other papers declare *subclasses* (cf. INROnto), we assume they mean the same.

(Duque-Ramos et al., 2013; Duque-Ramos et al., 2014) use the same metric, not for subclasses but superclasses. They describe the metric as the *"Average number of the direct **superclasses** per class minus the subclasses of Thing"* The recent papers (Franco et al., 2020; Manuel Quesada-Martínez et al., 2017), the wiki (*OQuaRE Wiki*, 2016), as well as the tool calculation (Fernandez-Breis et al., 2018) use the first published definition, but subtract the leaf classes: *"Number of the direct **subclasses** divided by the number of classes minus the number of **leaf classes**"*

We propose to use the metric by (Duque-Ramos et al., 2011; Duque-Ramos et al., 2016; *OQuaRE: A SQuaRE Based Quality Evaluation Framework for Ontologies;* M. Quesada-Martínez et al., 2015). At first, because it represents the most commonly cited definition. Secondly because the name alone suggests the use of subclass relationships.

2.2 RFCOnto (Response for a Class)

This metric is defined by (Duque-Ramos et al., 2013; Duque-Ramos et al., 2014) as *"Number of **Datatype** Properties and **Object Properties** that can be **directly accessed** from the class"*. (Duque-Ramos et al., 2011; *OQuaRE: A SQuaRE Based Quality Evaluation Framework for Ontologies)* not only state object and data properties but declare that it is the *"Number of **properties** that can be directly accessed from the Class"*. Even though this definition would include annotation properties, later versions state annotation properties explicitly. We, thus, assume that the intention from the second definition does not differ from the previous. (Duque-Ramos et al., 2016; M. Quesada-Martínez et al., 2015) describe RFCOnto as the *"Number of **usages** of **object** and **data properties** and **superclasses** divided by the number of classes **minus** the **subclasses** of **Thing**"*.

The recent papers (Franco et al., 2020; Manuel Quesada-Martínez et al., 2017) dropped the subtraction of the subclasses of thing, otherwise stating the same: *"Number of usages of **object** and **data properties** and **superclasses** divided by the **number** of **classes**"*. The wiki and the tool both implemented the latest calculation methodology (Fernandez-Breis et al., 2018; *OQuaRE Wiki*, 2016).

For this metric, we chose the widely used definition (Duque-Ramos et al., 2011; Duque-Ramos et al., 2013; Duque-Ramos et al., 2014; *OQuaRE: A SQuaRE Based Quality Evaluation Framework for Ontologies)* that includes the subtraction of the subclasses of the root class.

2.3 RROnto (Relationship Richness)

A first heterogeneity is in the naming: (Duque-Ramos et al., 2011; *OQuaRE: A SQuaRE Based Quality Evaluation Framework for Ontologies)* define RROnto as *Property Richness*. (Duque-Ramos et al., 2013; Duque-Ramos et al., 2014; *OQuaRE Wiki*, 2016) describe the metric as PROnto, *Property Richness*. Afterward, the naming is reverted to the abbreviation RROnto and the meaning *Relationship Richness*. While there exists another metric called PROnto, it was first introduced metric-wise in 2017 by (Manuel Quesada-Martínez et al., 2017) and is further utilized by (Franco et al., 2020). The wiki and the tool (Fernandez-Breis et al., 2018; *OQuaRE Wiki*, 2016) also describe PROnto but swapped the meaning of RROnto and PROnto. To untangle the confusion for the different naming conventions, we use the newest term *RROnto* and *Relationship Richness* for this metric.

Regarding metric definitions, the first paper and documentation (Duque-Ramos et al., 2011; *OQuaRE: A SQuaRE Based Quality Evaluation Framework for Ontologies)* describe RROnto as the *"Number of **properties** defined in the ontology divided by the number of relationships and properties"*. (Duque-Ramos et al., 2013; Duque-Ramos et al., 2014; Duque-Ramos et al., 2016; Manuel Quesada-Martínez et al., 2017) describes the metric differently as *"Number of usages of **object** and **data properties** divided by the number of **subclassof** relationships and properties"*. The formula presented in the online documentation (*OQuaRE: A SQuaRE Based Quality Evaluation Framework for Ontologies)* specifies that the definition uses the number of properties per class (object property assertions), not the object properties defined in the ontology generally. The terminology of relationship in the paper (Duque-Ramos et al., 2011) is the same as in NOCOnto and other metrics and is seen as equivalent to subclass relationships. Thus, we can assume that the two definitions describe the same thing.

The newest publication (Franco et al., 2020) exchanges the subclassof to a superclass relationship (which is equivalent) and puts it into the dividend: *"Number of usages of object and data properties and super classes divided by the number of classes"*. The tool does not follow the paper definitions and calculates it as the number of used properties divided by the sum of the used properties and the direct ancestor classes.

Again, we select the most cited metric, defined by (Duque-Ramos et al., 2011; Duque-Ramos et al., 2013; Duque-Ramos et al., 2014; Duque-Ramos et al., 2016; *OQuaRE: A SQuaRE Based Quality Evaluation Framework for Ontologies;* M. Quesada-Martínez et al., 2015; Manuel Quesada-Martínez et al., 2017).

2.4 PROnto (Properties Richness)

At first, the definitions for PROnto seemed very diverse. However, as shown for the RROnto metric, the name PROnto was sometimes assigned for the metric RROnto. PROnto in its distinctive form was first introduced by the papers (Franco et al., 2020; Manuel Quesada-Martínez et al., 2017) as the: *"Number of **subclassof** relationships **divided** by the number of **subclassof** relationships and **properties**"*. The wiki and the tool implement this definition but swapped the names of PROnto and RROnto

(Fernandez-Breis et al., 2018; *OQuaRE Wiki*, 2016). PROnto is, thus, not inconsistent in the definitions but the naming. We suggest using the PROnto for the definition named above.

2.5 TMOnto (Tangledness)

The definitions of tangledness are not as widespread as some of the other metrics. The first paper (Duque-Ramos et al., 2011) and the documentation (*OQuaRE: A SQuaRE Based Quality Evaluation Framework for Ontologies)* define it as the *"Mean number of **parents** per **class**"*. The papers (Duque-Ramos et al., 2016; Franco et al., 2020; M. Quesada-Martínez et al., 2015; Manuel Quesada-Martínez et al., 2017) define it as the *"Mean number of **classes** with **more** than **1** direct **ancestor**"*. We choose the older definition, as it is more broadly accepted, having more than six times the citations compared to the newer papers.

2.6 WMCOnto (Weighted Method Count)

The metric stays relatively consistent throughout the first five papers and the documentation. The paper published in 2011, 2015, and 2016 (Duque-Ramos et al., 2011; Duque-Ramos et al., 2016; *OQuaRE: A SQuaRE Based Quality Evaluation Framework for Ontologies;* M. Quesada-Martínez et al., 2015) define it as *"Mean number of properties and relationships per class"*, the ones published in 2013 and 2014 (Duque-Ramos et al., 2013; Duque-Ramos et al., 2014) declare it as the *"mean (2013 paper: average) number of Datatype Properties, Object Properties and **subclasses** per class"*. As in NOCOnto and INROnto, we assume that "relationships" are equivalent to subclass declarations.

The latest papers, the wiki, and application (Fernandez-Breis et al., 2018; Franco et al., 2020; *OQuaRE Wiki*, 2016; Manuel Quesada-Martínez et al., 2017) shift heavily in their meaning of WMCOnto and define it as the *"Mean **length** of the **path** from thing to a (!sic) leaf classes.* However, we suggest using the older, often cited version.

2.7 AROnto (Attribute Richness)

The challenge with *AROnto* is not the heterogeneity of its definition; it is defined as by (Duque-Ramos et al., 2011; Duque-Ramos et al., 2014; Duque-Ramos et al., 2016; Franco et al., 2020; M. Quesada-Martínez et al., 2015; Manuel Quesada-Martínez et al., 2017) as the *"number of **restrictions** of the ontology per classes"*, (Duque-Ramos et al., 2013; *OQuaRE: A SQuaRE Based Quality Evaluation Framework for Ontologies)* define it as the *"mean number of **attributes** per class"*, we assume that they both mean the same thing. However, the meaning of restriction or attribute in the context of ontologies is not fully clear if it is not concerned with properties (as the analysis of properties is already covered in other metrics). An implementation effort (Tibaut, 2018), thus, interpreted the metric as the number of property restrictions (*owl:someValuesFrom, owlhasvalue, ...*) divided by the number of classes.

(Manuel Quesada-Martínez et al., 2017) give more insights into the meaning of AROnto and describes it as "the number of elements that can be related by properties". This metric is implemented by the tool as well (Fernandez-Breis et al., 2018). The tool takes into account the *domain* axioms of *data* and *object properties* and counts how many classes (including subclasses) can be linked with the given properties. We adopted this calculation method.

3 HARMONIZED OQuaRE FRAMEWORK

As we have shown in the previous section, the available OQuaRE papers seem inconsistent in some of their proposed metrics. An imprecise terminology further contributes to the fuzziness in the metric definitions. The following section targets to translate the proposed metrics to a precise notation, clear heterogeneities, and build a joint base for future implementations of the OQuaRE metrics.

At first, we present the homogenized metric calculation. Subsection two is concerned with the collected quality characteristics. Subsection three recapitulates the metric interpretations that are part of OQuaRE.

3.1 Metric Definitions

The metrics are at the core of the OQuaRE framework. The following two tables collect the harmonized metrics. At first, Table 1 introduces the fundamental ontology attributes that the OQuaRE metrics build on. Every measurement is connected to a distinct symbol and comes with an example. Table 2 then presents the harmonized OQuaRE metrics, using the symbols previously introduced.

The metrics presented below are either homogeneously described in the OQuaRE publications or previously discussed in section two.

Table 1: Ontology Attributes Needed for Calculating the OQuaRE Metrics.

Symbol	Meaning
c_i	The i^{th} class of the ontology *E.g.: Class "Mother"*
$a_i(O)$	Annotation i on ontology O, does not include $a_i(c)$ *E.g.: This ontology is about family relations*
$a_i(c)$	Annotation i on class c *E.g.: Mother, Description: A Mother is a female who has at least one child*
$ind_i(c)$	Individual i of a class c *E.g.: Karen is instanceOf Mother*
$sub_i(c)$	Subclass i of the class c *E.g.: Mother is subClassOf Parent*
$dp_i(c)$	Data property assertion i on class c. *E.g.: Person subClassOf (Age exactly 1 sxd:integer)*
$op_i(c)$	Object Property assertion i on class c *E.g.: Daughter isRelativeOf some Mother*
$dom_i(DP)$	Classes in the domain i of data property DP (incl all subclasses) *E.g.: Age Domain Person*
$dom_i(OP)$	Classes in the domain i of object property OP (incl all subclasses) *E.g.:isRelativeOf Domain Person*
DP_i	Data property i declared in ontology *E.g.: Age*
OP_i	Object property i declared in ontology *E.g.: isRelativeOf*
$sup_i(c)$	Superclass i of the class c *Parent superClassOf Mother*
$root$	Root class of ontology *E.g., owl:thing*
$leaf_i$	Leaf class i, a leaf class does not have a subclass. *E.g.: Mother*
$\overrightarrow{path_i(c)}$	Path i from $root$ to c. *E.g.: root → Parent → Mother*
$\lvert\overrightarrow{path_i(c)}\rvert$	Length of a path i from $root$ to class c *E.g.: |root → Parent → Mother | = 3*

3.2 OQuaRE Quality Characteristics

On top of the OQuaRE metrics, OQuaRE defines the quality model. It provides desirable ontology features based on the software evaluation framework SQuaRE. It comprises high-level quality characteristics, each having further sub-characteristics. Parts of the quality characteristics are described in (Duque-Ramos et al., 2011; Duque-Ramos et al., 2013; Duque-Ramos et al., 2016; M. Quesada-Martínez et al., 2015). The data is available to a full extent on the OQuaRE webpage and wiki (*OQuaRE Wiki*, 2016; *OQuaRE: A SQuaRE Based Quality Evaluation Framework for Ontologies)*. The first published quality model, extensively described in (*OQuaRE: A SQuaRE Based Quality Evaluation Framework for Ontologies)* and used by (Duque-Ramos et al., 2011; Duque-Ramos et al., 2013), comprises 51 sub characteristics. The wiki, referenced by the rest of the papers, dropped two characteristics *C8-reliability* (3 sub characteristics) and *C9-performance efficiency* (2 sub characteristics), and split two up (C2.6, 2.7 & C.2.12, C2.13), resulting in a total of 48 quality characteristics.

Table 2: OQuaRE Metrics.

Metric	Formula
ANOnto *Annotation richness*	$\frac{\sum_{j,i} a_j(c_i) + \sum_i a_i(O)}{\sum_i c_i}$
AROnto *Attribute richness*	$\frac{\sum_{j,i} dom_j(DP_i) + \sum_{j,i} dom_j(OP_i)}{\sum_i c_i}$
CBOnto *Coupling between objects*	$\frac{\sum_i \sup(c_i)}{\sum_i c_i - sub(root)}$
CROnto *Class richness*	$\frac{\sum_{j,i} ind_j\,(c_i)}{\sum_i c_i}$
DITOnto *Depth of subsumption hierarchy*	$\max_i \lvert\overrightarrow{path}\rvert\,(c_i)$
INROnto *Relationships per class*	$\frac{\sum_{j,i} sub_j(c_i)}{\sum_i c_i}$
NACOnto *Number of ancestor classes*	$\frac{\sum_{j,i} sup_j(leaf_i)}{\sum_i leaf_i}$
NOCOnto *Number of children*	$\frac{\sum_{j,i} sub_j(c_i)}{\sum_i c_i - \sum_j sub_j(root)}$
NOMOnto *Number of properties*	$\frac{\sum_{j,i} dp_j(c_i) + \sum_{j,i} op_j(ci_i)}{\sum_i c_i}$
LCOMOnto *Lack of cohesion in methods*	$\frac{\sum_i \lvert\overrightarrow{path}\rvert(leaf_i)}{\sum_i leaf_i}$
RFCOnto *Response for a class*	$\frac{\sum_{j,i} dp_j(c_i) + \sum_{j,i} op_j(ci_i) + \sum_{j,i} sub_j(ci_i)}{\sum_i c_i - \sum_i sub_i(root)}$
RROnto *Relationship richness*	$\frac{\sum_{j,i} dp_j(c_i) + \sum_{j,i} op_j(ci_i)}{\sum_{j,i} sub_j(c_i) + \sum_{j,i} dp_j(c_i) + \sum_{j,i} op_j(ci_i)}$
TMOnto *Tangledness*	$\frac{\sum_{j,i} sup_j\,(c_i)}{\sum_i c_i}; \{\Sigma_j\, sup_j(c) > 1\}$
TMOnto2 *Tangledness 2*	$\frac{\sum_{k,j,i} sup_k\,(sup_j(c_i))}{\sum_{j,i} sup_j\,(c_i)}; \{\Sigma_j\, sup_j(c) > 1\}$
WMCOnto *Weighted method count*	$\frac{\sum_{j,i} dp_j(c_i) + \sum_{j,i} op_j(ci_i) + \sum_{j,i} sub_j(ci_i)}{\sum_i c_i}$
WMCOnto2 *Weighted method count 2*	$\frac{\sum_{j,i} \overrightarrow{path_j}(leaf_i)}{\sum_i leaf_i}$
PROnto *Property richness*	$\frac{\sum_{j,i} sup_j(c_i)}{\sum_{j,i} dp_j(c_i) + \sum_{j,i} op_j(ci_i) + \sum_{j,i} sub_j(ci_i)}$
Ponto *Ancestors per class*	$\frac{\sum_{j,i} sup_j(c_i)}{\sum_j c_i}$

The elements presented below are the union out of (*OQuaRE Wiki*, 2016; *OQuaRE: A SQuaRE Based Quality Evaluation Framework for Ontologies)* and the papers (Duque-Ramos et al., 2011; Duque-Ramos et al., 2013; Duque-Ramos et al., 2016; M. Quesada-Martínez et al., 2015), resulting in a total of 53 sub characteristics. The descriptions of the characteristics presented in the tables below are shortened and refined. The ordering of the item encapsulates no further meaning.

The first characteristic, "**structural**", evaluates the connections within an ontology and the attributes of the graph.

Table 3: Sub-Characteristics for Characteristic "Structural".

#	Sub-Characteristic	Description
C1.1	Formalization	The ontology is built on top of a formal model (*e.g., OWL, OBO*) to support reasoning
C1.2	Formal Relations Support	The ontology supports formal relations beyond taxonom
C1.3	Redundancy	All knowledge items are informative
C1.4	Structural Accuracy	The terms are correct
C1.5	Consistency	No set of items are contradictory or conflicting
C1.6	Tangledness	The fewer multi inheritance relationships are declared, the better
C1.7	Cycles	The existence of cycles is usually bad design and shall be avoided
C1.8	Cohesion	The classes are strongly related
C1.9	Domain Coverage	The ontology covers the specified domain (*requires an expert evaluation*)

Functional Adequacy describes the capability to provide concrete functions. Regarding this characteristic, the two documentations have minor differences. In (*OQuaRE: A SQuaRE Based Quality Evaluation Framework for Ontologies)*, the elements *Clustering* and *Similarity* are fused, as well as *Guidance*, and *Decision Trees*.

Table 4: Sub-Characteristics for Characteristic "Functional Adequacy".

#	Sub-Characteristic	Description
C2.1	Reference Ontology	The ontology can be used as a reference source
C2.2	Controlled Vocabulary	Heterogeneity is avoided. The ontology provides terminology management (*e.g., through the use of labels*)
C2.3	Schema and Value Reconciliation	The ontology provides a common data model and integrations to achieve semantic interoperability
C2.4	Consistent search and query	The formal model and structure guides the search process for data by providing concepts, machine-computable properties, and axioms
C2.5	Knowledge Acquisition	The capability of the ontology to represent the knowledge acquired in the form of instances
C2.6	Clustering	The annotations of terms enable clustering
C2.7	Similarity	The components can be compared for (*e.g., taxonomy, relation, attribute, or semantic*) similarity
C2.8	Indexing and Linking	The classes can act as indexes for fast information retrieval
C2.9	Result Representation	The ontologies capability to analyze complex results
C2.10	Classifying Instances	Instances can be recognized as class members with defined properties
C2.11	Text Analysis	Structure supports association detection between words and concepts to classify word types
C2.12	Guidance	Capability to guide the specification of domain theories through capturing knowledge and constraints about a domain
C2.13	Decision Trees	Capability to build decision trees
C2.14	Knowledge Reuse	The knowledge base can be used to build other ontologies
C2.15	Inference	The capability to use reasoners to make implicit knowledge explicit
C2.16	Precision	The ontology provides the right results with the needed accuracy

Compatibility describes the ability of at least two software components to exchange information and/or to perform their required functions while sharing a hardware or software environment.

Table 5: Sub-Characteristics for Characteristic "Compatibility".

#	Sub-Characteristic	Description
C3.1	Replaceabililty	The ontology can replace another ontology with the same purpose in the same environment
C3.2	Interoperability	The ontology can cooperatively combine its knowledge with other ontologies

Transferability describes the degree to which software can be transferred from one environment to another.

Table 6: Sub-Characteristics for Characteristic "Transferability".

#	Sub-Characteristic	Description
C4.1	Portability	The ontology or parts of it can be transferred between environments
C4.2	Adaptability	The ontology can be adapted to different specified environments *(e.g., languages, expressivity levels)*

Operability is concerned with the effort that is needed to use the ontology by stated or implied users.

Table 7: Sub-Characteristics for Characteristic "Operability".

#	Sub-Characteristic	Description
C5.1	Appropriateness Recognisability	The ontology enables the users to detect faults
C5.2	Learnability	The ontology enables users to learn its applications
C5.3	Ease of Use	It is easy for the users to operate and control the ontology
C5.4	Helpfulness	The application assists the users

Maintainability describes the capability of ontologies to be modified for changing environments, requirements, or functional specifications.

Table 8: Sub-Characteristics for Characteristic "Maintainability".

#	Sub-Characteristic	Description
C6.1	Modularity	The ontology is composed of discrete components. Changing one has minimal effect on the others
C6.2	Reusability	A part of the ontology can be used in other ontologies
C6.3	Analyzability	The ontology can be diagnosed regarding deficiencies, inconsistencies
C6.4	Changeability	The ontology can be easily modified
C6.5	Modification Stability	Unexpected effects from modifications are avoided
C6.6	Testability	The ontology can be validated

The characteristic **Quality in Use** measures how a product used by specific users meets their needs to achieve their goals. It is the quality in a particular context of use. Unlike the other quality criteria, this characteristic does describe additional sub characteristics.

Table 9: Sub-Characteristics for Characteristic "Quality in Use".

#	Sub-Characteristic	Description
C7.1.1	Usability in Use → Effectiveness in Use	A specified user can achieve their goals with accuracy and completeness in their context of use
C7.1.2	Usability in Use → Efficiency in Use	The used resources match the ontologies effectiveness
C7.1.3	Usability in Use → Satisfaction in Use	The user are satisfied in their specified context of use
C7.1.3.1	Usability in Use → Likability	Cognitive satisfaction
C7.1.3.2	Usability in Use → Pleasure	Emotional satisfaction
C7.1.3.3	Usability in Use → Comfort	Physical satisfaction
C7.1.3.4	Usability in Use → Trust	*Not further described*
C7.2.1	Flexibility in Use → Context Conformity in Use	Usability in use meets requirements of the intended context of use
C7.2.2	Flexibility in Use → Context Extendibility in Use	Usability in use in a context beyond initially intended

The following two quality characteristics are part of the first version of the quality framework published in (*OQuaRE: A SQuaRE Based Quality Evaluation Framework for Ontologies)*. They were later dropped in the wiki (*OQuaRE Wiki*, 2016).

Performance Efficiency describes the relationship between software performance and resource consumption under stated conditions.

Table 10: Sub-Characteristics for Characteristic "Performance Efficiency".

#	Sub-Characteristic	Description
C8.1	Response Time	The ontology provides appropriate response times and throughput rates
C8.2	Resource Utilization	The application uses the appropriate amount and types of resources when using the ontology

Reliability is concerned with maintaining the level of performance under the stated conditions.

Table 11: Sub-Characteristics for Characteristic "Reliability".

#	Sub-Characteristic	Description
C9.1	Error Detection	The ontology enables the users to detect faults
C9.2	Recoverability	The ontology can re-establish a specified performance level/data recovery in case of a failure
C9.3	Availability	The software component (*language, tools, ontology*) is operational and available when needed

4 CONNECTING QUALITY CHARACTERISTICS AND METRICS

As already stated earlier, the unique feature of OQuaRE is the holistic view on quality. Not only quality metrics are proposed, but also an interpretation of which values are desirable. Further, (Duque-Ramos et al., 2013) and the OQuaRE wiki (*OQuaRE Wiki*, 2016) state how quality metrics influence the quality characteristics shown in the section above. The section on quality influences in the online documentation (*OQuaRE: A SQuaRE Based Quality Evaluation Framework for Ontologies)* is no longer accessible. The collected information of the paper and the wiki is presented in Table 12.

By analyzing the results, one can see that some of the metrics are not described as an influence on quality characteristics (cf., *PROnto*, *POnto*, *NACOnto*). All second versions of metrics (thus, end with a "*2"* like *TMOnto2*) are also not marked as influencing a quality characteristic. However, as they are concerned with similar aspects like their first versions, we assume they also influence the same quality characteristics (e.g., *TMOnto2* also influences C1.6, C2.13).

Furthermore, while some metrics are connected to just two metrics *(TMOnto, CROnto)*, others have much more diverse connections, like *AROnto*, *NOMOnto* (associated with eight metrics), *RROnto*, and *WMCOnto* (associated with eight metrics).

Further, OQuaRE describes two influences on quality characteristics that are not associated with a metric but are itself a sub-characteristic of the category *Structural*: *Formality (C1.1)* and *Consistency (C1.5)*.

Table 12: Quality Interpretations for Metrics and their Influences on Quality Characteristics.

Metric	Influences	Best				Worst
		5	4	3	2	1
ANOnto	C1.3, C2.2, C2.4, C2.5, C2.6, C2.14	*>0,8*	*0,6-0,8*	*0,6-0,4*	*0,4-0,2*	*<0,2*
AROnto	C2.3, C2.4, C2.7, C2.8, C2.9, C2.12, C2.13, C2.14	*>0,8*	*0,6-0,8*	*0,6-0,4*	*0,4-0,2*	*<0,2*
CBOnto	C4.2, C5.2, C6.1, C6.2, C6.3, C6.4, 6.5	*1-3*	*3-6*	*6-8*	*8-12*	*>12*
CROnto	C2.9, C2.15	*>0,8*	*0,6-0,8*	*0,6-0,4*	*0,4-0,2*	*<0,2*
DITOnto	C3.1, C4.2, C6.2, C6.3, C6.4	*1-2*	*2-4*	*4-6*	*6-8*	*>8*
INRonto	C2.4, C2.8, C2.12, C2.13, C2.14	*>0,8*	*0,6-0,8*	*0,6-0,4*	*0,4-0,2*	*<0,2*
NACOnto		*1-2*	*2-4*	*4-6*	*6-8*	*>8*
NOCOnto	C3.1, C4.2, C5.2, C6.2, C6.4, C6.5	*1-2*	*2-4*	*4-6*	*6-8*	*>8*
NOMOnto	C2.5, C2.14, C3.1, C4.2, C5.2, C6.2, C6.3, C6.4	*<=2*	*2-4*	*4-6*	*6-8*	*>8*
LCOMOnto	C1.8, C2.14, C5.2, C6.3, C6.4, C6.5	*<=2*	*2-4*	*4-6*	*6-8*	*>8*
RFCOnto	C4.2, C5.2, C6.2, C6.3, C6.4, C6.5	*1-3*	*3-6*	*6-8*	*8-12*	*>12*
RROnto	C1.2, C2.3, C2.4, C2.5, C2.7, C2.8, C2.15	*>0,8*	*0,6-0,8*	*0,6-0,4*	*0,4-0,2*	*<0,2*
TMOnto	C1.6, C2.13	*1-2*	*2-4*	*4-6*	*6-8*	*>8*
TMOnto2		*1-2*	*2-4*	*4-6*	*6-8*	*>8*
WMCOnto	C3.1, C4.2, C5.2, C6.1, C6.2, C6.3, C6.4	*1-2*	*2-4*	*4-6*	*6-8*	*>8*
WMCOnto2		*1-2*	*2-4*	*4-6*	*6-8*	*>8*
PRONTO		*>0,8*	*0,6-0,8*	*0,6-0,4*	*0,4-0,2*	*<0,2*
PONTO		*>0,8*	*0,6-0,8*	*0,6-0,4*	*0,4-0,2*	*<0,2*
Formality	C2.2, C2.3, C2.4, C2.11, C2.14, C2.15					
Consistency	C2.2, C2.3, C2.14					

While we argue that there are metrics available that could be used to measure these two aspects, e.g., *Richness* and *Lawfulness* by Burton-Jones et al. (Burton-Jones et al., 2005), or "*Meta-Logical Adequacy"* and "*Generic Complexity"* by Gangemi et al. (Gangemi et al., 2005), the definition of new OQuaRE-metrics is beyond the scope of this paper. Further, it is noted that we did not carry out an analysis of whether stated connections between quality characteristics and metrics or the proposed metric quality ranges are valid, there are merely collected out of the various resources.

5 CONCLUSION & OUTLOOK

There are many different aspects one can consider when evaluating ontologies. Ontology quality frameworks can guide which metrics to use and how to interpret them. Furthermore, from the proposed frameworks, OQuaRE probably provides the most holistic assessment and guidance. However, as shown in this paper, the various published papers and resources on OQuaRE sometimes seem to contradict each other. The heterogeneities make the implementation of the framework difficult.

The presented paper aims at providing a single permanent point of reference for future use of the framework. We homogenized the proposed metrics and provided a clear, formalized description of the metrics. The various sources of OQuaRE are consolidated, making it the first peer-reviewed paper that shows OQuaRE to its full extent.

Harmonizing the OQuaRE quality framework is one next step in a broader effort to research the practical use of ontology quality frameworks using evolutional data (Reiz, 2020). In the upcoming months, we are going to build a software tool to analyze large amounts of ontologies using various automatic ontology quality frameworks. Here, we aim to learn what makes a good ontology and how the proposed frameworks measure the right aspects. We believe that this research also has the potential to validate the stated connections between metric and quality characteristics and metric value recommendations.

REFERENCES

Burton-Jones, A., Storey, V. C., Sugumaran, V., & Ahluwalia, P. (2005). A semiotic metrics suite for assessing the quality of ontologies. *Data & Knowledge Engineering*, *55*(1), 84–102. https://doi.org/10.1016/j.datak.2004.11.010

Duque-Ramos, A [Astrid], Boeker, M., Jansen, L., Schulz, S., Iniesta, M., & Fernández-Breis, J. T [Jesualdo Tomás] (2014). Evaluating the Good Ontology Design Guideline (GoodOD) with the ontology quality requirements and evaluation method and metrics (OQuaRE). *PloS One*, *9*(8), e104463. https://doi.org/10.1371/journal.pone.0104463

Duque-Ramos, A [Astrid], Fernández-Breis, J. T [Jesualdo Tomás], Iniesta, M., Dumontier, M., Egaña Aranguren, M., Schulz, S., Aussenac-Gilles, N., & Stevens, R [Robert] (2013). Evaluation of the OQuaRE framework for ontology quality. *Expert Systems with Applications*, *40*(7), 2696–2703. https://doi.org/10.1016/j.eswa.2012.11.004

Duque-Ramos, A [Astrid], Fernández-Breis, J. T [J. T.], Stevens, R [R.], & Aussenac-Gilles, N. (2011). OQuaRE: A square-based approach for evaluating the quality of ontologies. *Journal of Research and Practice in Information Technology*, *43*(2), 159–176. https://www.scopus.com/inward/record.uri?eid=2-s2.0-84860428632&partnerID=40&md5=dcb393aa78ee79eca9bfe365b38ed0f1

Duque-Ramos, A [Astrid], Quesada-Martínez, M [Manuel], Iniesta-Moreno, M., Fernández-Breis, J. T [Jesualdo Tomás], & Stevens, R [Robert] (2016). Supporting the analysis of ontology evolution processes through the combination of static and dynamic scaling functions in OQuaRE. *Journal of Biomedical Semantics*, *7*(1), 63. https://doi.org/10.1186/s13326-016-0091-z

Fernandez-Breis, J. T., Franco, M., Vivo, J. M., Quesada-Martinez, M., Duque-Ramos, A [A.], & Bernabe-Diaz, J. A. (2018). *OQuaRE Calculation Web Service* [Computer software]. Universidad de Murcia. http://sele.inf.um.es/ontology-metrics

Franco, M., Vivo, J. M., Quesada-Martínez, M [Manuel], Duque-Ramos, A [Astrid], & Fernández-Breis, J. T [Jesualdo Tomás] (2020). Evaluation of ontology structural metrics based on public repository data. *Briefings in Bioinformatics*, *21*(2), 473–485. https://doi.org/10.1093/bib/bbz009

Gangemi, A., Catenacci, C., Ciaramita, M., Lehmann, J., Gil, R., Bolici, F., & Strignano Onofrio. (2005). *Ontology evaluation and validation: An integrated formal model for the quality diagnostic task.*

OQuaRE Wiki. (2016, November 8). http://miuras.inf.um.es/oquarewiki

OQuaRE: A SQuaRE based Quality evaluation framework for Ontologies. http://miuras.inf.um.es/evaluation/oquare/

Quesada-Martínez, M [M.], Duque-Ramos, A [A.], & Fernández-Breis, J. T [J. T.] (2015). Analysis of the evolution of ontologies using OQuaRE: Application to EDAM. *CEUR Workshop Proceedings*, *1515*. https://www.scopus.com/inward/record.uri?eid=2-s2.0-84960969476&partnerID=40&md5=47f57e4d8749dadeecee72b922fc853a

Quesada-Martínez, M [Manuel], Duque-Ramos, A [Astrid], Iniesta-Moreno, M., & Fernández-Breis, J. T [Jesualdo Tomás]. (2017). Preliminary Analysis of the OBO Foundry Ontologies and Their Evolution Using OQuaRE. In R. Randell, R. Cornet, C. McCowan, N. Peek, & P. J. Scott (Eds.), *Studies in health technology and informatics: volume 235. Informatics for health: Connected citizen-led wellness and population health* (pp. 426–430). IOS Press. https://doi.org/10.3233/978-1-61499-753-5-426

Reiz, A. (2020). An Evolutional Based Data-Driven Quality Model for Ontologies. In H. Alani & E. Simperl (Chairs), *ISWC-DC,* Athens, Greece/online. http://ceur-ws.org/Vol-2798/paper1.pdf

Reiz, A., Dibowski, H., Sandkuhl, K., & Lantow, B. (2020, November 2–4). Ontology Metrics as a Service (OMaaS). In *Proceedings of the 12th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management* (pp. 250–257). SCITEPRESS - Science and Technology Publications. https://doi.org/10.5220/0010144002500257

Tartir, S., Arpinar, I. B., Moore, M., Sheth, A. P., & Aleman-Meza, B. (2005). OntoQA: Metric-Based Ontology Quality Analysis. In *IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources,* Houston.

Tibaut, A. (2018, March 29). *ontology-evaluation*. https://freesoft.dev/program/124565495

Yao, H., Orme, A. M., & Etzkorn, L. (2005). Cohesion Metrics for Ontology Design and Application. *Journal of Computer Science*, *1*(1), 107–113. https://doi.org/10.3844/jcssp.2005.107.113