Semantic Web 14 (2023) 1051–1097
DOI 10.3233/SW-233393
IOS Press

A conceptual model for ontology quality assessment

A systematic review

R.S.I. Wilson [a,b,*], J.S. Goonetillake [b], W.A. Indika [c] and Athula Ginige [d]

[a] *Department of Computer Science and Informatics, Uva Wellassa University, Badulla, Sri Lanka*
[b] *Department of Information Systems Engineering, University of Colombo School of Computing, Colombo 07, Sri Lanka*
[c] *Department of Computer Science, University of Ruhuna, Matara, Sri Lanka*
[d] *School of Computer, Data and Mathematical Sciences, Western Sydney University, NSW, Australia*

Editor: Aldo Gangemi, University of Bologna, Italy
Solicited reviews: Stefano De Giorgis, University of Bologna, Italy; Aldo Gangemi, University of Bologna, Italy; two anonymous reviewers

Abstract. With the continuous advancement of methods, tools, and techniques in ontology development, ontologies have emerged in various fields such as machine learning, robotics, biomedical informatics, agricultural informatics, crowdsourcing, database management, and the Internet of Things. Nevertheless, the nonexistence of a universally agreed methodology for specifying and evaluating the quality of an ontology hinders the success of ontology-based systems in such fields as the quality of each component is required for the overall quality of a system and in turn impacts the usability in use. Moreover, a number of anomalies in definitions of ontology quality concepts are visible, and in addition to that, the ontology quality assessment is limited only to a certain set of characteristics in practice even though some other significant characteristics have to be considered for the specified use-case. Thus, in this research, a comprehensive analysis was performed to uncover the existing contributions specifically on ontology quality models, characteristics, and the associated measures of these characteristics. Consequently, the characteristics identified through this review were classified with the associated aspects of the ontology evaluation space. Furthermore, the formalized definitions for each quality characteristic are provided through this study from the ontological perspective based on the accepted theories and standards. Additionally, a thorough analysis of the extent to which the existing works have covered the quality evaluation aspects is presented and the areas further to be investigated are outlined.

Keywords: Ontology quality, ontology quality model, ontology quality characteristics, systematic review

1. Introduction

An ontology is a formal representation of the concepts in a particular domain of knowledge and the relationships between those concepts [61]. More specifically, an ontology has been defined as “a formal, explicit specification of a shared conceptualization” in [123] by merging the two prominent definitions of Gruber [61]: “ontology is an explicit specification of a conceptualization” and Borster [21]: “an ontology is a formal specification of a shared

[*] Corresponding author. E-mail: shyama@uwu.ac.lk.

conceptualization". The *conceptualization* denotes an abstract and simplified view of the world that consists of concepts that are expected in the world being represented, and relationships among them [121]. Thus, ontology has also been defined as a representational vocabulary [61,103]. A *formal specification* is machine-interpretability in a way that a machine can understand and process the specified conceptualization. This would enable ontology to be used for several purposes such as to make meaningful communication among their agents/users, facilitate interoperability, knowledge sharing & reuse among subsystems and, to distinguish the domain knowledge from operational knowledge [103]. These can be viewed as capabilities of ontologies [103,138] and consequently, ontologies are increasingly incorporated into information systems.

However, the above capabilities would not be a reality unless a good quality ontology is produced at the end of the development. This is because one trivial internal quality issue would cause multiple quality issues in the end product which may result in a high cost of debugging and fixing the issues and in turn loss of consumers' trust. For instance, it has been revealed that the ontology-based system in the medicine domain has caused a 55% loss of information due to one missing explicit definition in the ontology [56,57,60,89,151]. Thus, it is required to sufficiently evaluate both the quality of the content of the ontology during the development, as well as, the ontology as a whole after the development [56]. This has been viewed as two aspects of ontology evaluation in [101,105,154] namely intrinsic and extrinsic that have been further discussed in Section 3. Moreover, it has been revealed that the availability of a widely accepted methodology or model, or approach for the quality evaluation of ontologies would be beneficial for producing a quality ontology [105]. However, such a methodology, model, or approach has not yet been identified for ontology quality evaluation as the way it is in the other related fields such as system and software engineering. For instance, the availability of widely agreed quality standards in software engineering such as ISO/IEC 9126 [78], ISO/IEC 9241-11 [76], ISO/IEC 25012 [75], ISO/IEC 25010 [77], and IEEE software quality standards [74] provide immense support in developing good quality software.

Since this article is on the ontology quality aspect, first we examine quality from the general perspective. Quality is referred to as the fitness of a product or service or information for the needs of the intended users in a specified context [31]. Basically, the quality of a system indicates how far the system caters to user needs with respect to a particular context. The context (i.e., the context of use) defines the actual conditions under which a product (i.e., software/Information Systems (IS)) is used by intended users in performing a task (see Fig. 1). The context not only considers the technical environment of the system but also the user type (i.e., attitude, literacy, experience), the physical and social environment, the available equipment and tasks to be performed [31,127].

Quality characteristics are a set of properties that can be used to assess whether the intended *user needs* of a system or product are met. In software and system engineering, there are a set of standards [76–78] namely; ISO/IEC 2501n, ISO/IEC 2501n and ISO/IEC 2503n [77], which describe how the *quality characteristics* to be evaluated are identified with respect to the *user needs* of a system. According to the standards, the *quality requirements*,

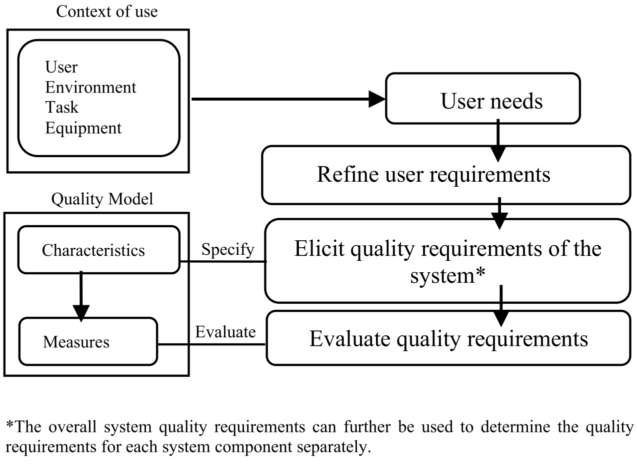

Context of use
User
Environment
Task
Equipment
User needs
Refine user requirements
Quality Model
Characteristics
Measures
Specify
Evaluate
Elicit quality requirements of the system*
Evaluate quality requirements
*The overall system quality requirements can further be used to determine the quality requirements for each system component separately.

Fig. 1. The abstraction process of specifying quality requirements.

which can also be viewed as a set of *quality characteristics*, should be elicited from *user needs* at the requirement specification phase [17,141] (see Fig. 1). To this end, a *quality model* plays a significant role that consists of a proper set of quality *characteristics* and associated *measures*. The *quality model* provides a framework for determining *characteristics* in relation to *user needs* [75,77,127]. The identified *quality characteristics* can be evaluated at the intrinsic and extrinsic levels of a product using their respective *measures*. This evaluation helps to ensure that the required quality is being achieved across product development. Figure 1 shows the abstraction process of deriving the *quality requirements* and how a *quality model* supports that process. The significance of a *quality model* is further elaborated in Section 3 with an example.

If we apply the process in Fig. 1 to the ontology context, it is possible to define the *user* as a person who interacts with an ontology-based IS. *Context of use* is the user type (as stated above), the environments (social, physical, technical) that the users may involve and the task to be performed through the ontology. *User needs* are the set of *requirements* that are expected to be fulfilled by the ontology-based IS. Additionally, the *quality requirements* are a set of *overall quality requirements** for the ontology-based IS elicited from the *user needs* in a particular *context of use*. From these overall *quality requirements*, it is possible to derive the *quality requirements* related to *the ontology* in a system. Thereafter, the *ontology quality requirements* can be specified using *quality characteristic*s with the support of an *ontology quality model*. This concept further discusses in detail in Section 3 (see Fig. 5).

It can be understood that quality models play a significant role in determining the quality characteristics of a system. When considering the ontological point of view, there are several quality models which have been proposed for ontology quality assessments. However, none of them is widely accepted in practice. This could be because some of the models such as the semiotic metric suit [24], the quality model of Gangemi et al. [48,49], and OQuaRE [40] are generic. Thus, it is required to additional efforts to customize and add absent characteristics which are significant to the specified context. Moreover, some of the quality models such as OntoQualitas [114] and the quality model of Zhu et al. [110] are specific to a domain (i.e., domain-dependent). Therefore, these models do not guarantee to work well for domains other than their specified domain. Consequently, the researchers and practitioners face the following difficulties.

(i) Difficulty in determining the required set of quality characteristics and, in turn, the relevant quality measures to achieve the specified quality requirements [105].
(ii) Difficulty in differentiating the quality characteristics and quality measures as their definitions in the literature are vague and the terminologies have been used interchangeably [147]. This also has a negative impact on (i) above.
(iii) Inadequate knowledge in assessing the complete set of quality characteristics applicable for an ontology since the ontology quality assessment in practice has got limited only to a certain set of characteristics irrespective of many other characteristics which would become applicable and have been theoretically proposed [32].

Thus, there is in fact a requirement to perform a thorough analysis on the domain of ontology quality models and problems. As a result, a systematic review was performed to streamline the findings of the existing quality problems and thereby produce a conceptual quality model that is useful for researchers and practitioners to understand the characteristics, attributes, and measures to be considered in different aspects of ontology quality assessment. Moreover, in ontology development, ontology can be developed entirely from scratch or can be reused from the existing ontologies or a combination of both. However, ontology quality assessment is performed against the same set of requirements (i.e., characteristics) specified in a particular context regardless of whether the developed ontology is built from scratch or reused [102]. Thus, the conceptual model proposed in this study is helpful for assessing the quality of ontologies irrespective of how they are developed (i.e., from scratch and/or reused). Furthermore, when discussing the conceptual quality model, we focus on web ontologies which are built upon one of the standard web ontology languages such as RDF(S) [136] and different variants of OWL [135]. However, some quality characteristics such as compliance, consistency, completeness and accuracy can also be associated with other ontology languages. Moreover, we do not distinguish between TBox (i.e., ontology structure) and ABox (i.e., knowledge base) when discussing the quality characteristics. The term ontology is used to refer to both TBox and ABox.

The rest of the article is structured as follows; Section 2 presents the survey methods that we have followed. Section 3 provides preliminaries and conceptualization of ontology quality assessment to understand the concepts that have been described in the rest of the sections. The overview of the existing ontology quality models is presented in Section 4. In Section 5, the quality characteristics, associated quality measures, and definitions for each quality characteristic are described. Section 6 provides a discussion of the findings of the survey. Finally, Section 7 discusses the important gaps related to ontology quality models/approaches and Section 8 concludes the survey with highlights of future works.

2. Survey methodology

Firstly, a traditional theoretical review was performed by retrieving the related ontology quality surveys and the literature reviews, to explore the ontology quality evaluation and then, to identify the possible quality models, including significant ontology quality characteristics and measures, and the relations among the characteristics. Few attempts have been carried out to present such contributions [51,93,108,112,140]. However, we consider that they are not comprehensive surveys as they have not clearly defined: the research gaps and questions that they have addressed and the survey methodology that they have followed. Furthermore, these surveys have provided only a general overview. Moreover, the discussions were limited to a set of characteristics proposed in [48,57,58,62,142]. In addition to that, few comprehensive surveys on ontology evaluation have been identified (see Table 1) in which the main focus is on ontology evaluation in broader aspects, not specifically for ontology quality problems.

We conducted the systematic review by following the methodologies described in [23,83] with the objective of addressing the issues (i), (ii), and (iii) stated in Section 1. Thus, the review includes a summarization of ontology quality problems and models: recognition of characteristics and measures. Eventually, the aim is to present a conceptual model that provides a basis for researchers to retrieve the quality characteristics upon the quality requirements. To achieve these objectives, the following research questions were derived;

- What ontology quality models are proposed for ontology quality assessment?
- What quality problems are discussed in the previous approaches with respect to ontology quality?
- What are the ontology quality characteristics and measures assessed in the previous approaches?

Table 1

The existing survey studies in ontology evaluation

Article	Description
Vrandecic, 2010 [142]	**Databases**: Not specified **Scope**: ontologies specified in Web Ontology languages **Results**: Presented the overview of domain- and task-independent evaluation of an ontology by emphasizing the quality assessment related to the aspects: vocabulary, syntax, structure, semantics, representation, and context.
Gurk et al, 2017 [95]	**Databases**: Three databases: ScienceDirect, IEEE Xplore Digital Library, ACM Digital Library (the selected articles have not been mentioned) **Scope**: ontologies specified in Web Ontology languages **Results**: Performed a survey to find the ontology quality metrics for a set of characteristics that are defined in ISO/IEC 25012 Data Quality Standard (This work is limited only to the inherent quality and inherent-system quality)
Degbelo, 2017 [32]	**Databases**: Semantic Web Journal, Journal of Web Semantics (2003-2017) **Scope**: ontologies specified in Web Ontology languages **Results**: Performed the review on quality criteria and strategies which have been used in the design and implementation stages of ontology development. Presented the gaps between theory and practice of ontology evaluation.
McDaniel and Storey, 2019 [96]	**Databases**: Web of Science journals, approximately 170 articles **Scope**: ontologies specified in Web Ontology languages **Results**: Performed the review on the evaluation of domain ontology. Classified and discussed the existing ontology evaluation research studies under the five classes: Domain/Task fit, Error-checking, Libraries, Metrics, and Modularization.

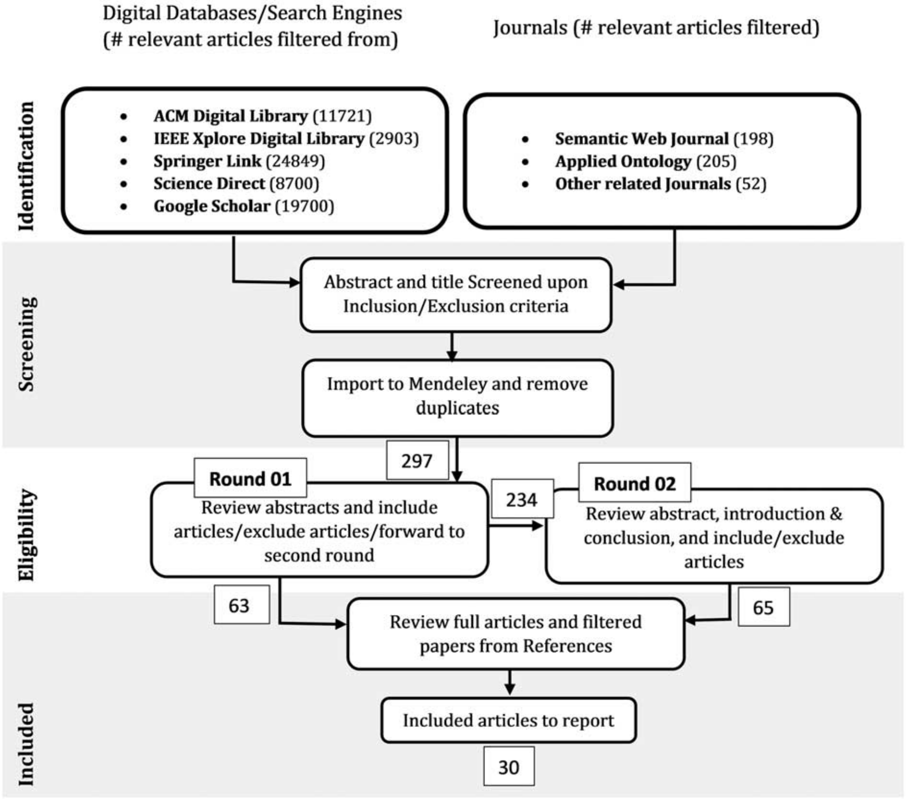

Digital Databases/Search Engines
(# relevant articles filtered from)
Journals (# relevant articles filtered)
Identification
ACM Digital Library (11721)
IEEE Xplore Digital Library (2903)
Springer Link (24849)
Science Direct (8700)
Google Scholar (19700)
Semantic Web Journal (198)
Applied Ontology (205)
Other related Journals (52)
Screening
Abstract and title Screened upon Inclusion/Exclusion criteria
Import to Mendeley and remove duplicates
297
Eligibility
Round 01
Review abstracts and include articles/exclude articles/forward to second round
234
Round 02
Review abstract, introduction & conclusion, and include/exclude articles
63
65
Review full articles and filtered papers from References
Included
Included articles to report
30

Fig. 2. Main steps of the review process and number of articles retrieved.

Then, the search terms were identified based on the research questions such as *quality*, *ontology*, *ontology quality*, *assessment*, *evaluation*, *approach*, *criteria*, *measures*, *metrics*, *attributes*, *characteristics*, *methods, and methodology*, and trial searches were executed using various combinations of the terms. Thereafter, the following combination was used to retrieve the relevant papers from the selected digital databases, journals, and search engines (see Fig. 2).

– [Ontology AND [Quality OR Evaluation OR Assessment]]
– [[Ontology AND Quality] AND [Criteria OR Measures OR Metric OR Characteristics OR Attributes]]
– [Ontology AND Quality] AND [Models OR Approach OR Methodology]]

These search combinations were performed according to the instructions given in each digital database under the advanced search.

The following inclusion and exclusion criteria were defined; to reduce the likelihood of publication bias, to select the recently discussed studies and to keep the sample size down to make the review easier to handle. Accordingly, the defined inclusion and exclusion criteria were used to select the candidate studies at the screening and eligibility phases (see Fig. 2),

– Inclusion Criteria: *Studies that are in English, published during the period* (2010–2021), *were peer-reviewed. Studies discussed ontology quality assessment, empirical or theoretical studies*.
– Exclusion Criteria: *Studies published as a short paper* (*less than 6 pages*), *tutorial, poster, and report. Studies did not present a rigorous approach to achieve the defined quality assessment objectives*. *Studies do not directly relevant to the research questions*.

During the screening phase, the candidate studies were retrieved by performing a keyword search on titles and abstracts. Then, the duplicate studies were removed using the reference management tool (i.e., Mendeley). Thereafter, the abstracts were reviewed to filter the relevant studies. There are some cases wherein the abstract is vague and difficult to understand the real contribution. In this case, the papers were forwarded to the second round for further review of the introduction and conclusion as suggested in [23]. Altogether, 128 papers were forwarded for the full paper review. During this stage, forward and backward citations were searched and included the appropriate studies to report. It has been identified that some papers have two versions: journal and conference entitled to the same authors. In such a situation, the journal paper was selected. Finally, thirty (30) papers mentioned in the Appendix were selected for the reporting phase and they have been considered as the core papers in the survey. In addition to that, the milestone papers, which were captured through the backward citation searching (i.e., before 2010), have also been taken into account for the discussions (see the Appendix).

3. Preliminaries and conceptualization

In the context of quality, a number of ad hoc definitions and inconsistent terminologies appear in the literature leading to terminology misapplications and misinterpretations [13]. Thereby, in this section, the concepts and the terms that are relevant to our discussion are briefed to avoid miscommunication and thus to maintain consistency. Additionally, the conceptualization of ontology quality assessment is provided based on the existing theories and Table 2 presents a list of terms used in this article.

Terminologies related to quality models Quality models provide the basis for specifying quality requirements and evaluating the quality of an entity (i.e., software, tools, ontology, part of the software) [75,77]. In software engineering, quality models are twofold, relational (i.e., non-hierarchical) and hierarchical [55]. The relational model presents the correlation among the quality characteristics and this type of model has not been proposed for the ontology so far in the existing works. Only, the hierarchical models can be found which have usually four levels (see Fig. 3). The top-level (i.e., the first level) consists of a set of characteristics that are further decomposed into sub-characteristics at the next level (i.e., the second level). The third level would consist of associated attributes of characteristics/sub-characteristics. These attributes have measures, which lay at the bottom level (i.e., the fourth level), that can be used to assess the entity either quantitively or qualitatively. Thus;

Characteristics (i.e., Criteria): describe a set of attributes. An attribute is a measurable physical or abstract property of an entity [84]. In the ontology quality context, an entity can be a set of concepts, properties, or an ontology [80].

Measure (i.e., metrics): describes an attribute formally and assesses them either quantitively or qualitatively [11].

Table 2

Related terms and definitions

Term	Definition
Quality model	Describes a set of characteristics and the relationships between them, also provides the basis for specifying quality requirements and evaluating the quality of an entity [75,77]
Context of use	Users, tasks, equipment (hardware, software and materials), and the physical and social environments in which a product is used [75,77]
Quality in use	The degree to which a product or system can be used by specific users to meet their needs to achieve specific goals with effectiveness, efficiency, freedom from risk and satisfaction in specific contexts of use [75,77]
Semiotics	The field of study of signs and their representations [107]
Design patterns	Reusable modeling solutions to recurrent ontology design problems [47]
Anti-pattern	Domain-related ontological misrepresentations [71]
Pitfalls	Potential errors, modeling flaws, and missing good practices in ontology development [111]
Lawfulness	Correctness of syntactic rules of the ontology profile [5,24]
Language Richness	The amount of ontology-related syntax features used [24]
Interpretability	Meaningfulness of terms [24]
Clarity	Comprehensibility of the term label [5,24]
Cohesion	The degree to which the elements of a module belong together [96]
Coupling	The degree of relatedness between ontology modules [104]
Tangledness	The multi-hierarchical nodes of a graph (i.e., a class with several parent classes) [50]

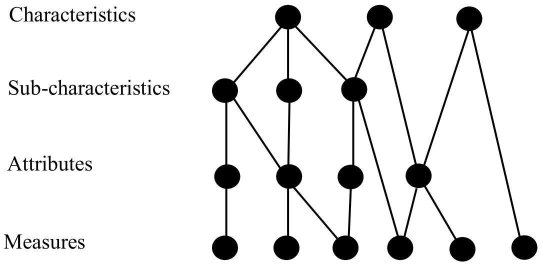


Fig. 3. Association between characteristics, attributes, and measures.

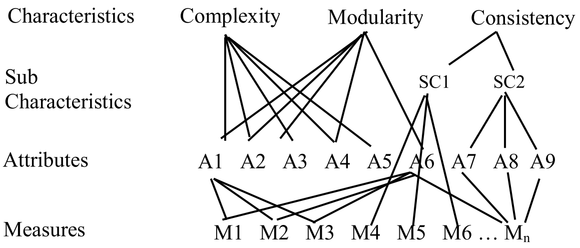


Fig. 4. An example for a simple quality model.

There are no definite requirements to have sub-characteristics for each characteristic or else characteristics/sub-characteristics should have attributes. Thereby, a characteristic can directly be related to measures without a set of attributes (see Fig. 3). Moreover, the same attributes/measures can be used to assess many characteristics. This has been further illustrated in Fig. 4 with the selected ontology characteristics. For instance, *complexity*, which is one of the ontology characteristics, describes the properties of the ontology structure (i.e., taxonomy, non-taxonomy) [48,49,116,142]. It is associated with several attributes such as *size* ($A1$), *depth* ($A2$), *breadth* ($A3$), *fan-outness* ($A4$) and *density* ($A5$) of an ontology [50] and each attribute has a set of measures. If the *size* attribute is considered, the related measures are *the number of classes* ($M1$), *the number of relations* ($M2$), and *the number of instances* ($M3$). Thus, the associated levels of *complexity* related to quality models are <characteristics, attributes, measures>.

Moreover, the measures of the attributes: *size, depth, breadth*, and *fan-outness* have also been used to assess the *cohesion* ($A6$) attribute of the *modularity* characteristic [132,153].

If we consider the *consistency* characteristic of an ontology, it can be further derived into two sub-characteristics namely, *internal consistency* (*SC1*) and *external consistency* (*SC2*) [58,153]. *Internal consistency* contains direct measures such as *circularity errors* ($M4$), *the number of multi-parent classes* ($M5$), and *the number of misused disjointness* ($M6$) [56,58]. Thus, with respect to that scenario, the associated levels in the quality model are <characteristics, sub-characteristics, measures>. However, *external consistency* can be measured through attributes *interpretability* ($A7$), *clarity* ($A8$), *and precision* ($A9$) [96,131]. Each attribute has a set of measures that have been presented in Table 8. Thus, it covers levels <characteristics, sub-characteristics, attributes, measures>.

Moreover, sub-characteristics also can be further derived into another set of sub-characteristics also called primitive characteristics [19] as the way it is in OQuaRE [40]. Additionally, it can be observed that some of the quality models consist of another higher level called *dimensions* or *factors* on top of the characteristics-level. It has been proposed to classify the characteristics/attributes which are related to similar aspects of the ontology. For instance, the quality model of Gangemi et al. [48,49] contains three dimensions namely; *structural*, *functional* and *usability-related*. Similarly, the quality model of Zhu et al. [153] has classified the characteristics under the *content*, *presentation*, and *usage* dimensions. However, there is no clear definition is given for the dimension in literature, thus, the term: dimension has been used interchangeably to define the term characteristics. For instance, in [146,152], the term: dimension is used to define characteristics. However, in our study, we use the terms in the order: *dimensions, characteristics, sub-characteristics, attributes*, and *measures* to describe the structure of a quality model whenever it is necessary.

Ontology layers (i.e., levels) The researchers have discussed the ontology quality layer-wise, or level-wise by concerning an ontology as a multi-layered vocabulary [22,57,108,112,142]. Initially, three layers to be focused on have been proposed in [57], namely: *content, syntactic & lexicon*, and *architecture*. Later, this was expanded by including the layers: *hierarchical* and *context* [22,142]. The *syntactic layer* considers the properties related to the language that is used to represent the ontology formally [22,57]. The *hierarchy* or *taxonomy* layer focuses on the properties related to the taxonomic structure (i.e., *is-a* relationship) of ontologies [22]. A number of measures have been defined related to the taxonomy layer [42,48,50,54,116]. These measures are often useful to observe the extent to which the concepts in an ontology are spread out in relation to the root concept of that ontology [42,116] and to track the evolution of ontologies easily [142]. The authors in [42,116] have shown that taxonomic measures such as *maximum depth/breadth, average depth* and *depth/breadth variance* are good predictors of ontology *reliability*. The *architectural layer* is known as the *structural* or *design* layer in [22,108]. It considers whether an ontology is modeled based on the pre-defined design principles and criteria [22,57]. The *lexicon layer* is also named as *vocabulary* or *data* layer [22,142], which takes into account the vocabulary that is used to describe concepts, properties, instances, and facts. The non-hierarchical relationships and semantic elements are considered under the *semantic* layer. To evaluate the *vocabulary*, *architecture* and *semantic* layers, some understanding of domain knowledge is required. The *context* layer concerns the application scope that the ontology is built for. Mainly, it determines whether the ontology satisfies the application requirements as a component of an information system or a part of a collection of ontologies [22]. We have performed a separate analysis on the ontology evaluation layers and their relationship between the evaluation approaches (i.e., methods): *task-based, human-based, data-driven*, and *golden-based*, which can be found in [147] for more detail.

Ontology development methodologies and design patterns Ontology methodologies and design patterns and principles provide guidelines for knowledge modeling and representation paving the path for accurate knowledge capturing and defining. This enables ontology developers to achieve the quality characteristics emphasized in this study such as compliance, consistency, completeness, and conciseness. Due to this reason, it is necessary to explore the development methodologies, design patterns and principles. However, discussing them in detail is out of the scope of this paper.

There are several *methodologies* introduced for ontology development such as Cyc [92], Uschold and King's method [139], TOVE [64], METHONTOLOGY [59], SENSUS [128], On-To-Knowledge [120] and DILIGENT [110]. Among them, only Uschold and King's method, TOVE and On-To-Knowledge have discussed *ontology evaluation* as a phase to be performed after the development of the entire ontology. The METHONTOLOGY mechanism performs the evaluation at every phase of ontology development. However, none of these methodologies have neither described nor revealed the ontology evaluation methods in detail [27,69]. More details of the comparison of these methodologies can be found in [27,29,87]. In addition to that, agile ontology development methodologies have been introduced by adopting agile principles and practices in software engineering. These include methodologies such as SAMOD [109], XD [18,30], AMOD [1] and CD-OAM framework [129]. Out of these, only SAMOD and XD discussed the evaluation (i.e., model test, data test, unit test) of ontologies at least in a nutshell. The significance of XD (i.e., eXtreme Design with Content Ontology Design Patterns) is the use of ontology content design patterns to construct ontologies that instinctively support to prevent of making common modeling mistakes. This, in turn, directs ontologists to construct a quality ontology, particularly assisting inexperienced ontologists [18].

Ontology Design Patterns (ODPs) are reusable modeling solutions to recurrent ontology design problems [47]. Several groups have proposed a set of design patterns. For instance, the authors in [6] have proposed seventeen ODPs in relation to the biological knowledge domain. They have classified ODPs into three groups namely (i) *extension ODPs* which provide solutions to bypass the limitations of OWL such as n-ary relations, (ii) *good practice ODPs* which support to construct of robust and cleaner design (iii) *domain modeling ODPs* which provide solutions for concrete modeling problems in biology [6]. Moreover, the *semantic web best practices and deployment* working group has introduced ODPs which include n-ary relations, classes as property values, value partitions/sets, and simple part-whole relations. In addition to that, the group ontologydesignpatterns.org suggests more than two hundred patterns and principles which have been categorized into six families namely; content ODPs, structural ODPs, correspondence ODPs, reasoning ODPs, presentation ODPs and lexico-syntactic ODPs [9]. The reuse of the patterns and principles can be determined at the ontology design phase with respect to the requirements of the ontology development and it makes ontology modeling faster [71,102]. To this end, the authors in [18,71] have shown through experiments that pattern-based ontology development avoids making common modeling mistakes. This in turn improves the quality and usability of ontologies.

Furthermore, OntoClean has proposed a set of principles (i.e., rules) that helps to identify problematic modeling choices associated with taxonomy relationships [65]. Mainly, the rules of OntoClean have been constructed upon a set of metaproperties related to taxonomy namely; identity, unity, dependence, essence and rigidity. By embedding these principles, an *ontology-driven conceptual modeling language* has been proposed namely OntoUML [16,71,106]. Moreover, OntoUML provides a set of design patterns that accelerate ontology modeling. In addition to that, it describes a set of anti-patterns to be avoided in modeling ontologies.

Ontology evaluation space An ontology-based information system usually consists of many components. The quality of each component of the system would influence the overall quality and in turn usability of the system which is often critical for business success [17,35]. As explained in Section 1, the quality requirements which are to be achieved through a system or each system component should be traced from user needs also known as business requirements (see Fig. 1). Specification of quality requirements can be viewed as a connected flow starting from user needs to the internal quality of a system. For instance, *quality requirements* can be elicited from *user needs* and they can also be defined as *a set of quality requirements* to be achieved by a system when it is in use. This is also defined as *quality in use* (see Fig. 5) [17,75]. Then, this *set of quality requirements* can be used to determine *the external quality requirements* of each component of a system, accordingly, determines *the internal quality requirements*. These quality requirements (i.e., Internal/External) can also be expressed as a set of *characteristics* with associated *measures*. For that, a quality model can be used that provides a framework for determining characteristics/sub-characteristics in relation to the *quality requirements*. Then, these characteristics/sub-characteristics can be evaluated using the relevant measures to ensure whether the desired quality in each stage is achieved (see Fig. 5).

For an ontology-based system, ontology quality requirements also need to be distinctly identified from the *overall quality requirements* of the system [141]. Based on that, the external and internal quality requirements to be achieved from the ontology can be derived. The external and internal quality have been discussed as *extrinsic* and *intrinsic* evaluation aspects of ontologies in [101,105]. Figure 5 presents the flow of specifying the ontology quality requirements with the associated evaluation aspects when an ontology is a component of a system [148]. It should

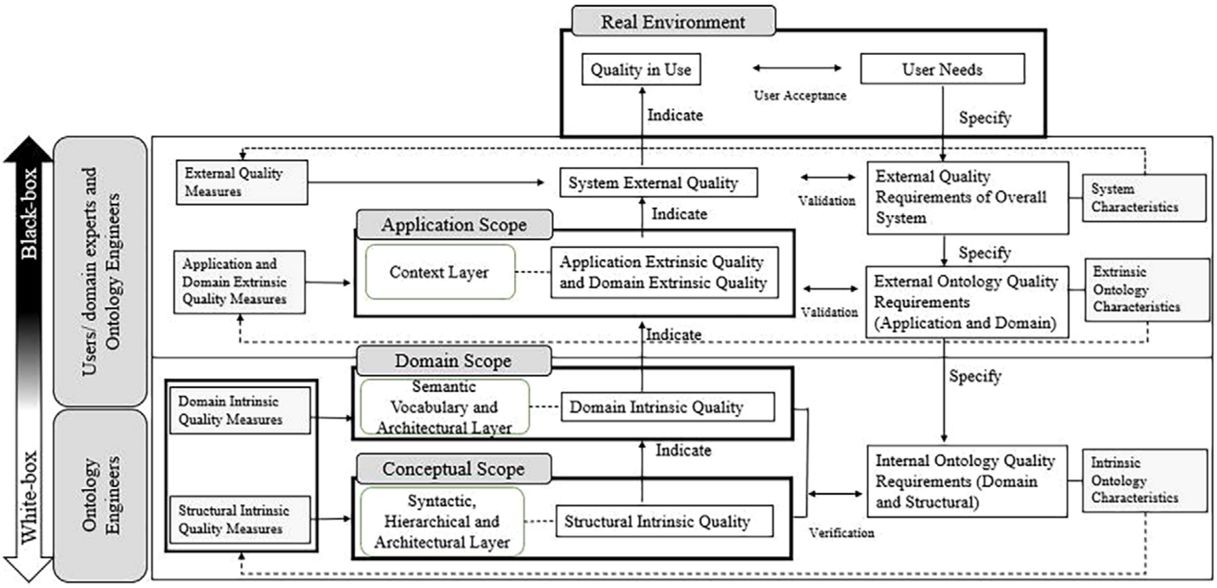

Real Environment
Quality in Use
User Acceptance
User Needs
Indicate
Specify
Black-box
White-box
Users/ domain experts and Ontology Engineers
Ontology Engineers
External Quality Measures
System External Quality
Validation
External Quality Requirements of Overall System
System Characteristics
Indicate
Application Scope
Specify
Application and Domain Extrinsic Quality Measures
Context Layer
Application Extrinsic Quality and Domain Extrinsic Quality
Validation
External Ontology Quality Requirements (Application and Domain)
Extrinsic Ontology Characteristics
Indicate
Domain Scope
Specify
Domain Intrinsic Quality Measures
Semantic Vocabulary and Architectural Layer
Domain Intrinsic Quality
Indicate
Conceptual Scope
Internal Ontology Quality Requirements (Domain and Structural)
Intrinsic Ontology Characteristics
Structural Intrinsic Quality Measures
Syntactic, Hierarchical and Architectural Layer
Structural Intrinsic Quality
Verification

Fig. 5. Conceptualization of ontology quality evaluation [148].

be noted that the quality requirements related to the other software and hardware components of the ontology-based system have not been presented in Fig. 5.

The extrinsic evaluation considers quality under two aspects namely *domain extrinsic* and *application extrinsic*. The ontology *quality requirements* elicited from the *user needs* can be separated under these aspects [125,141]. For instance, quality requirements associated with domain knowledge are considered under the domain extrinsic aspect. On the other hand, ontology quality requirements that are specifically needed by the system that the ontology is deployed are considered under the application extrinsic aspect [101,105]. Moreover, the application extrinsic requirements are independent of the domain knowledge. In performing a quality evaluation under these aspects, an ontology is taken into account as a component of a system. Traditionally this evaluation is called "black-box" or "Task-based" testing [70,101] as the quality assessment being performed without peering into the ontology structure and content. Furthermore, this extrinsic evaluation has been defined as ontology validation in [121]. Accordingly, ontology is assessed to confirm whether the right ontology was built [121,142], mainly this is performed with the involvement of users and domain experts (see Fig. 5). Moreover, this has also been considered as the context-layer evaluation in [22,142].

The intrinsic evaluation focuses on the ontology content and structural quality to be achieved in relation to the extrinsic quality requirements. To this point, ontology is considered as an isolated component separated from the system. At this stage, ontology quality can be evaluated under two aspects: *domain intrinsic* and *structural intrinsic* [101], mainly performed by ontology developers. In the structural intrinsic aspect, the focus is given to the syntactic and structural requirements (i.e., syntactic, structural, architectural layers) which involve the specified conceptualization such as language compliance, conceptual complexity, and logical consistency. In the domain intrinsic aspect, ontology quality is evaluated with reference to the domain knowledge that is required for the intended needs of users [101]. Thus, ontology developers need to get the assistance of domain experts to evaluate this aspect. Moreover, the semantic, vocabulary and architectural layers of ontologies are evaluated under this aspect. Furthermore, the quality evaluation performed under the structural and domain intrinsic aspects are similar to the paradigm of "white-box" testing in software engineering [70]. Thus, from the intrinsic aspects, verification is being done to ensure whether the ontology is built in the right way [121,142].

The quality requirements identified under each aspect of the ontology evaluation space can be associated with the corresponding set of characteristics together with measures. Then, those characteristics can be measured in order to ensure whether the defined quality requirements are being achieved through ontology development. In the following section, we illustrate how the quality characteristics can be derived in relation to an ontology requirement using a use case in agriculture.

Use case We considered an ontology-based decision support system that provides pest and disease management knowledge for farmers. For simplicity, we assumed that ontology quality requirements have already been elicited from user needs and as such consider that the requirement specified below is the key quality requirement of the system.

– The ontology should provide correct pest and disease knowledge for user queries requested through the system.

From the extrinsic aspect of the ontology, the mentioned quality requirement is associated with domain knowledge in agriculture. Thus, it is a domain extrinsic quality requirement and it discusses the *accuracy* of knowledge to be provided through the ontology. At this stage, the *accuracy* of the knowledge can be observed in terms of the accuracy of answers given for the competency questions [150]. Moreover, *accuracy* should be evaluated with the assistance of domain experts and users. With respect to the identified extrinsic quality requirement, it is required to define what set of characteristics should be met from the intrinsic aspect. For that, we need to utilize techniques such as ROMEO [150] and GQM [11] that support deriving the intrinsic quality requirements from the higher-level (i.e., extrinsic) quality requirements. By adopting the ROMEO methodology [150], the following questions are formulated to derive the intrinsic quality requirements. For instance, to provide accurate pest and disease knowledge for farmers, domain knowledge should be correctly modeled for an ontology. For that, all required axioms[1] with regard to pest and disease management should be correctly defined in the ontology. Accordingly, the question: $Q1$ can be derived. In addition to that, the ontology should be free from internal contradiction to reason and produce the correct knowledge. Thus, the question: $Q2$ also can be derived with respect to the defined extrinsic ontology requirements. It is noteworthy that we have not described all the steps followed under the ROMEO methodology and only the derived questions are highlighted. However, an interested reader may refer to the article [148,150].

- Q1: Does the ontology capture the axioms correctly to represent knowledge in pest and disease management?
- Q2: Is the ontology free from internal contradiction?

Based on the derived questions in relation to the extrinsic quality requirement, it is required to identify the associated characteristics and the relevant measures to be evaluated. In this situation, ontology quality models play a key role that supports identifying the corresponding characteristic/sub-characteristic in relation to the derived questions. However, there is no such a well-formed quality model for ontologies as highlighted in Section 1. Thus, to this end, we determined the characteristics based on the literature survey. For instance, Q1 discusses *external consistency* as it considers whether the defined axioms are consistent with the domain knowledge. Q2 discusses *internal consistency* as it considers whether there are any internal contradictions within the ontology definitions. After identifying the characteristics, the relevant measures to be evaluated through ontology development should be identified. This process would be easier if a well-formed quality model is available. For the use case that we have considered, *external consistency* can be evaluated using the measure of *precision*, as explained in [150], which provides a ratio between the correctly defined axioms ($\mathrm{O_i}$) and the total axioms defined in the ontology ($\mathrm{O_a}$). The correctness/consistency of axioms ($\mathrm{O_i}$) can be determined against a frame of reference ($\mathrm{F_a}$) such as a valid corpus or the standard ontology [150]. Precision values can range from zero (0) to one (1). If the value of precision is zero (0) then it implies that the particular ontology has not correctly captured any of the axioms defined in the frame of reference. If the value of precision is 0.6 then it implies that the ontology has correctly captured 60% of the axioms with respect to the frame of reference.

If the value of precision is one (01) then it implies that the particular ontology has correctly captured all axioms with respect to the considered frame of reference. Meantime, it can be stated that the particular ontology has correctly represented the relevant knowledge of the domain.

Moreover, *internal consistency* can be assessed by observing the number of logical inconsistencies with the support of a reasoner. In addition to that, the measures such as the number of circularity errors, the number of subclass partitions with common classes and the number of partitions with common instances can be observed (see Table 7). These measures are considered as a set of pitfalls that could lead to wrong inference [58,111,114] and they can be detected using the tool OOPS! [111].

In this way, for the rest of the ontology quality requirements, the relevant ontology characteristics can be determined with the support of a quality model. This use case is further explained in the article [148] to illustrate how the other ontology characteristics (i.e., coverage, comprehensibility) and their measures are derived for the ontology requirements. These characteristics can then be measured through ontology development to ensure that the ontology being modeled is of good quality.

4. Ontology quality models

4.1. *Quality models in software engineering*

It is worthwhile to analyze the related contribution in neighboring research fields that support identifying established theories and practices relevant to our study. Moreover, the related theories and practices can be reused or adopted rather than developing the theories and practices from scratch. Therefore, the quality models in the software

[1] The term “axiom” is used in this article to denote to Declaration|ClassAxiom|ObjectPropertyAxiom|DataPropertyAxiom|DatatypeDefinition|Assertion|AnnotationAxiom AND sub axioms come under each of these axioms. Example: OWL https://www.w3.org/TR/owl2-syntax/#Axioms.

Table 3

Comparison of software quality model

Model	Levels and Characteristics
McCall's quality model [25]	**Levels:** factor, characteristics, measures **Characteristics**: correctness, reliability, efficiency, integrity, usability, maintainability, testability, flexibility, portability, reusability and interoperability
Boehm's quality model [19]	**Levels:** primary, intermediate, primitive **Characteristics (intermediate)**: portability, reliability, Efficiency, usability (Human Engineering), testability, understandability and modifiability
Garvin's quality model [53]	**Level:** product quality characteristics **Characteristics:** performance, features, reliability, conformance, durability, serviceability, aesthetics, and perceived quality
ISO/IEC 25012 [75]	**Levels:** inherent data and system-dependent data quality characteristics **Characteristics:** accuracy, completeness, consistency, credibility, currentness, accessibility, compliance, confidentiality, efficiency, precision, traceability, understandability, availability, portability and recoverability
ISO/IEC 25010 [78]	**Levels:** Factors (Quality in use, external quality, internal quality), characteristics, sub-characteristics and measures **Characteristics:** functional suitability, performance efficiency, compatibility, usability, security, maintainability, portability, effectiveness in use, efficiency in use, satisfaction, free from risk, context coverage

engineering field were explored. Many well-accepted quality models are available in software engineering such as McCall's quality model [25], Boehm's quality model [19], Garvin's quality model [53], the data quality model: ISO/IEC 25012[75], and the ISO 9126 quality model [78] (see Table 3).

McCall's quality model [25] has proposed eleven quality factors (i.e., can also be considered as dimensions) from the product perspectives which further have been divided into a set of characteristics and measures. Thus, it provides a hierarchal model containing three levels. Boehm's quality model is a hierarchical model that is similar to McCall's model. However, it provides a broader range of characteristics without defining the measures (see Table 3). The top level of Boehm's quality model contains the characteristics: *As-is utility, Maintainability*, and *Portability*. Based on that, the intermediate and low levels describe the related sub-characteristics and primitive characteristics respectively.

Garvin's quality model has described eight quality dimensions (i.e., factors) specifically for product quality that can be adapted to software engineering. Moreover, the dimensions have been defined at the abstract level, and some of them are subjective, thus it is difficult to measure.

The data quality model ISO/IEC 25012 has defined fifteen characteristics that have to be taken into account when assessing the quality of data of products [75]. The characteristics further have been categorized into two types namely *inherent* data quality and *system-dependent* data quality. The *inherent* data quality describes the characteristics of data such as accuracy, completeness, consistency, credibility, and currentness which have the intrinsic potential to satisfy stated and implied needs. The rest of the characteristics such as accessibility, compliance, confidentiality, efficiency, precision, traceability, understandability, availability, portability, and recoverability, come under the *system-dependent* data quality that must be preserved within a computer system when data is used under specified conditions.

ISO 9126 model (ISO/IEC 9126:1991) is a widely accepted software product quality model that was produced in 1991 by the International Organization for Standardization (ISO). Later, it has been extended as ISO/IEC 9126-1:2001 which includes four parts:

- Part 1: Quality model
- Part 2: External Measures
- Part 3: Internal Measures
- Part 4: Quality in use Measures

Part 1: the quality model comprises with six characteristics related to internal and external software product quality which have further been decomposed into sub-characteristics. The measures of each characteristic/sub-characteristic have been defined under Parts: 2, 3, and 4 of the standard.

Furthermore, SQuaRE: ISO/IEC 25010 (**S**ystems and software **Qua**lity **R**equirements and **E**valuation): has been introduced by redesigning the ISO 9126 model with the ISO 14598 series of standards. The reason for redesigning the ISO 9126 model is that it contains issues as a result of the advancement of information technologies and the changes in its environment [78]. SQuaRE comprises the same set of characteristics that were defined in ISO 9126 with several amendments as described in [78].

From the ontological perspective, there are no agreed quality models as the way it is in software engineering. However, significant contributions have been made to the field and some quality models have been developed by adopting theories in software engineering. For instance, OQuaRE [40] and SemQuaRE [113] are quality models related to ontologies, constructed upon the theories described in system and software standards: SQuaRE-ISO/IEC 25010 [78]. Additionally, we realized that the findings of our survey can be classified as the way it has been done for data quality in ISO/IEC 25012. Moreover, it has been identified that some of the characteristics of the data quality model such as *credibility, timeliness, recoverability* and *availability* can be adopted for the ontology quality. This has been further discussed in Section 5. The subsequent section discusses the existing quality models for ontologies.

4.2. Quality models for ontologies

Initially, a framework has been proposed in [57] to verify that developers are building a correct ontology. The proposed framework consists of a set of characteristics namely *soundness, correctness, consistency, completeness, conciseness, expandability*, and *sensitiveness*, which have been adopted in later research works and developments [96,114,142,150]. Thereafter, significant methods [65,103,138] and tools such as OntoTrack [82], OntoClean [65], OntoQA [132], OntoMetric [90], SWOOP [82] have been proposed enabling quality assessment with several sets of characteristics mainly focusing on the intrinsic aspect. Afterward, several attempts have been taken to provide a generalized quality model, significantly, the semiotic metric suit [24], the quality model of Gangemi [48,49], and OQuaRE [40].

Semiotic metric suite [24] *Semiotic theory* [122] has been taken as the foundation for this model. According to [122], the *semiotic theory* is “the study of the interpretations of signs and norms” and it defines six levels namely syntactic, semantic, pragmatic, social, physical, and empiric. The semiotic metric suite has been developed by classifying a set of ontology quality attributes under the aforementioned levels, excluding physical and empiric levels. These aspects can also be considered as quality *dimensions*, which are as follows.

- **Syntactic**: Lawfulness, Richness
- **Semantic**: Interpretability, Consistency, Clarity
- **Pragmatic**: Comprehensiveness, Accuracy, Relevancy
- **Social**: Authority, History

Even though the authors have mentioned that this model has been constructed solely by focusing on the intrinsic aspect, the attributes: *accuracy*, *relevancy*, and all attributes in the *social* dimension take on an extrinsic nature. To the reason that, under the pragmatic and social levels, the ontology is assessed as a whole and measures are evaluated with reference to the domain experts’ knowledge [5], and external documents: usage logs, page ranking, details of external links with other ontologies [96]. Accordingly, it can be understood that the semiotic metric suite provides a set of quality attributes related to both intrinsic and extrinsic aspects of an ontology.

The quality model of Gangemi et al. [48,49] By considering an ontology as a semiotic object “*including graph objects, formal semantic spaces, conceptualizations, and annotation profiles*”, the three main dimensions: *structural*, *functional*, and *usability-related* have been proposed. The *structural* dimension consists of thirty-two measures related to the topological, logical, and meta-logical characteristics of an ontology [50]. Primarily, it covers syntactic and formal semantics. Under the *functional* dimension, the possible attributes: *precision*, *recall* (*i.e.*, *coverage*), and *accuracy* have been proposed to assess the conceptualization specified by the ontology with respect to the intended use. The measures of the *usability-related* dimension focus on the metadata (i.e., annotations) about the ontology and the elements related to the communication context. To this end, the attributes: *presence, amount, completeness*, and *reliability* have been described under three levels: *recognition* annotation, *efficiency* annotation, and *interfacing* annotation by Gangemi et al. [48,49].

OQuaRE [40] By adopting the standard SQuaRE (ISO/IEC 25000:2005) [77], the OQuaRE quality model has been proposed to evaluate ontology quality. It comprises the same set of characteristics defined in the standard ISO/IEC 25000:2005 such as *functional adequacy, reliability, operability, maintainability, compatibility*, and *transferability*. In addition to that, the *structural* characteristic has been included in the model to assess the inherent topological characteristics of an ontology. Furthermore, these characteristics have been decomposed into several sub-characteristics considering the ontological point of view including a set of associated measures which are available online.[2]

Based on our survey, it has been recognized that OQuaRE [40] has not considered the semantic features of an ontology such as coherency and coverage of the domain knowledge. These features are essential for ontologies to process meaningful interpretations. In addition to that, the authors in [114,142,153] have highlighted the issues related to OQuaRE. They have stated that the proposed sub-characteristics are subjective and are difficult to be applied in practice. Consequently, OQuaRE [40] has not appeared in the later research. However, noticeable studies are available related to the other two models: semiotic metric suit [24] and the quality model of Gangemi et al. [48,49].

OntoKeeper [5] and DoORS [98] are semiotic-metric-suite-driven initiatives. For instance, OntoKeeper has automated the semiotic metric suite by taking into account the intrinsic aspect of an ontology. DoORS is a web-based tool for evaluating and ranking ontologies. It has been developed by extending the semiotic metric suite [24] with a set of additional characteristics: *structure*, *precision* (i.e., instead of clarity), *adaptability*, *ease of use*, and *recognition*.

The quality model of Gangemi et al. [48,49] has been adopted in multiple studies [38,54,111,142]. OOPS! [111] has used that model to classify the proposed common pitfalls which can occur during ontology development. Furthermore, OOPS! has extended the dimensions to another level as presented in Table 4. The authors in [54] have used the structural measures defined in the model of Gangemi et al. [48,49] to evaluate the cognitive ergonomics of an ontology. Moreover, a theoretical model proposed in [142] contains eight characteristics: *Accuracy, Adaptability, Clarity, Completeness, Computational efficiency, Conciseness, Consistency*, and *Organizational fitness*, which have been derived through literature reviews including the set of characteristics proposed in Gangemi et al.'s list [48,49]. The quality model proposed in the ONTO-EVO^{A}L approach has been constructed mainly based on Gangemi et al.'s model that contains two dimensions: *content* and *usage* with six characteristics (see Table 4) [38].

In addition to that, few efforts have been made in designing quality models for a specific purpose namely; OntoQualitas [114], the quality model for semantic descriptions of web services [153], and SemQuaRE [113]. OntoQualitas provides a set of characteristics and related measures to evaluate the quality of ontologies which are built upon the purpose of exchanging information between heterogeneous systems. The ROMEO methodology [150] is the basis that OntoQualitas follows to derive the measures relating to the quality characteristics. The applicability of the characteristics in OntoQualitas to the intended context was empirically evaluated. Focusing on the context of semantic descriptions of web services, the authors in [153] have presented a quality model which comprises four levels namely: *aspects, attributes, factors*, and *metrics* (see Table 4). These levels can be considered as *dimensions, characteristics, sub-characteristics/attributes*, and *measures* respectively as per our definitions in Section 3.

SemQuaRE [113] quality model has not specifically been defined for ontologies. However, it is for assessing the quality of semantic technologies. For instance, *ontology engineering tools, Ontology matching tools, Reasoning systems, Semantic search tools, and Semantic web services*.

Notably, there are two strategies that have been followed in constructing models for software quality [39] namely; the *top-down approach* and the *bottom-up approach*. The *top-down approach* starts to construct the model from the characteristics and then decomposes them into other levels which contain sub-characteristics and attributes respectively. Finally, the corresponding measures for each characteristic/attribute are identified. In contrast to that, the *bottom-up* approach first observes the related measures which have been defined in the existing studies and classifies them until reaching the top-level characteristics. [39]. When considering the discussed models related to ontologies, only SemQuaRE [113] model has been constructed following the *bottom-up* approach. All the other models [24,38,40,48,98,111,114,153] have followed the *top-down* approach, in which, the characteristics related to ontologies were first determined, and from that, the possible sub-characteristics/attributes and measures were

[2] http://miuras.inf.um.es/evaluation/oquare/Metrics.html

Table 4

The existing quality models for ontologies identified through the survey

Model	Description	Dimensions	Characteristics and Sub-Characteristics/attributes
Quality model in ONTO-EVO^{A}L [38], 2010	**Structure**: Hierarchical model **Base model**: Gangemi et al.'s model **Purpose**: To assess the quality of evolving ontology **Approach**: Top-down approach: the characteristics have been derived from [50,132,142] **Evaluation**: Empirical evaluation has not been presented.	Content	Complexity Cohesion Conceptualization Semantic Richness Attribute Richness Inheritance Richness Abstraction
		Usage	Completeness Precision Recall Comprehension
OQuaRE [40], 2011	**Structure**: Hierarchical model **Base model**: The SQuaRE standard **Purpose**: To rank, select, compare and assess the ontologies **Approach**: Top-down approach **Evaluation**: Empirical evaluation has been performed on two applications: Ontologies of Units of Measurement[a] and Bio ontologies[b]	–	Structural Functional Adequacy Reliability Operability Maintainability Compatibility Transferability (More sub-characteristics of these are available online[c])

Table 4

(Continued)

Model	Description	Dimensions	Characteristics and Sub-Characteristics/attributes
OntoQualitas [114], 2014	**Structure**: Hierarchical model **Base model**: No specific model is used **Purposed**: To evaluate the quality of an ontology whose purpose is the information interchanges between heterogeneous systems **Approach**: Top-down approach: adopted the ROMEO methodology [150], then the criteria have been derived from [24,58,65] **Evaluation**: Empirical evaluation has been performed on ontologies of enterprises interchange Electronic Business Documents	–	Language conformance Completeness Conciseness Correctness Syntactic Correctness Semantic Correctness Representation Correctness Usefulness
Quality model in OOPS! [111], 2014	**Structure**: Hierarchical model **Base model**: the model of Gangemi et al. [48,49] **Purpose**: To classify the identified pitfalls **Approach**: Top-down approach **Evaluation:** the model of Gangemi et al. [48,49] has been extended to classify the pitfalls, thus, no evaluation has been provided.	Structural	Correctness Modeling Completeness Ontology Language conformance
		Functional	Requirement Completeness Content Adequacy
		Usability-related	Ontology Understanding Ontology Clarity

Table 4

(Continued)

Model	Description	Dimensions	Characteristics and Sub-Characteristics/attributes
Quality model of Zhu et al. [153], 2017	**Structure**: Hierarchical model **Base model**: No specific model is used **Purpose**: To evaluate ontology for Semantic Descriptions of Web Services **Approach**: Top-down approach **Evaluation:** Empirical evaluation has been performed on five ontologies which underlying of five web services in applications of reporting weather forecasts. The evaluation has been performed against the standard[d] weather ontology	Content	Correctness Internal Consistency External Consistency Compatibility Completeness Syntactic Completeness Semantic Completeness
		Presentation	Well-formedness Conciseness Non-redundancy Structural Complexity Size Relation Modularity Cohesion Coupling
		Usage	Applicability Definability Description Complexity Adaptability Tailorability Composability Extendibility Transformability Efficiency Search Efficiency Composition Efficiency Invocation Efficiency Comprehensibility

Table 4
(Continued)

Model	Description	Dimensions	Characteristics and Sub-Characteristics/attributes
Quality model of McDaniel et al. [98], 2018	**Structure**: Hierarchical model **Base model**: The semiotic metric suit **Purpose**: To rank ontologies **Approach**: Top-down approach **Evaluation:** empirical evaluation has been performed by selecting ontologies from the Bio Portal ontology repository	Syntactic	Lawfulness
			Richness
			Structure
		Semantic	Consistency
			Interpretability
			Precision
		Pragmatic	Accuracy
			Adaptability
			Comprehensiveness
			Ease of use
			Relevance
		Social	Authority
			History
			Recognition

[a] http://miuras.inf.um.es/evaluation/oquare/UOM/ContMetricsUOM.html
[b] http://miuras.inf.um.es/evaluation/oquare/CTO/MetricsBiOntology.html
[c] http://miuras.inf.um.es/evaluation/oquare/Contenido.html
[d] https://www.auto.tuwien.ac.at/downloads/thinkhome/

derived. The quality models for ontologies that we have explored through the survey are summarized in Table 4. All of them are hierarchical quality models having the structure: <characteristics, sub-characteristics, attributes, measures> and there is no significant work identified on constructing a non-hierarchical (i.e., relational) quality model that shows the correlation between characteristics.

In Table 5, the characteristics/attributes proposed in the ontology quality models in [38,98,114,153] have been mapped with the respective aspects (i.e., *structural intrinsic, domain intrinsic, extrinsic* (*domain/application*), and *quality in use*). However, the models: OQuaRE [40] and the quality model in OOPS! [111] has not been included in Table 5. To the reason that the OQuaRE [40] characteristics have been defined by considering the ontology as a software artifact. Thus, it is difficult to distinctly map their characteristics with the ontological evaluation aspects. When considering OOPS!, it has adopted an existing model to classify a set of pitfalls and does not specifically provide a set of characteristics concerning the quality requirements.

If we consider other quality models, OntoQualitas and the model in ONTOEVO$^{\mathrm{A}}$L have defined characteristics mostly related to the intrinsic extent. Of which, the characteristics: *completeness* (coverage), *conciseness* (*precision*), and *representational correctness*, are associated with the domain that the ontology is considered (see Table 5). Additionally, OntoQualitas has taken *quality in use* of ontology into account by defining the characteristics *usefulness*, i.e., *usefulness* of the ontology for the heterogeneous information interchange. The quality model of Zhu et al. [153] concerns both intrinsic and extrinsic aspects. However, the sub-characteristics of *applicability* and *efficiency* have been specifically defined for the semantic web services context. When comparing these models, the characteristics proposed in the quality model of McDaniel et al. [98] cover many aspects of ontology quality evaluation. In which, *authority*, *history*, and *recognition* reflect user satisfaction. For instance, *authority* considers the number of linkages with other ontologies, *history* considers the number of revisions made to the ontology and how long it has been actively public. *Recognition* considers the number of times the ontology is downloaded and the reviews given to the ontology. Thus, these attributes are useful to understand to what extent the ontology is accepted by the community, then the positive values of the attributes imply user satisfaction with the ontology.

In addition to that, by adopting the GQM (Goal-Question-Metrics) methodology [11], the study [150] employed in providing approaches to derive the measures of quality characteristics tracing from ontology requirements. In which, goals are the ontology requirements that are gradually refined into questions/sub-questions which reflect the respective quality characteristics to be measured. In this case, quality models act as a complementary component that supports deriving measures with respect to the characteristics reflected in each question. Thus, the proposed approaches would not be effective without a quality model that presents a set of characteristics and corresponding measures.

5. Classification of ontology quality characteristics

Several characteristics and measures have been discussed in the selected papers. From that, fourteen (14) significant characteristics: *compliance, complexity, internal consistency, modularity, conciseness, coverage, external consistency, comprehensibility* (*i.e.*, *intrinsic point of view*), *accuracy, relevancy, functional completeness, understandability* (*i.e.*, *extrinsic point of view*), *adaptability, efficiency*, were identified. Then, the identified characteristics were grouped under the four *evaluation aspects* which also can be defined as dimensions namely: *Structural intrinsic*, *Domain intrinsic*, *Domain extrinsic* and *Application extrinsic*. These dimensions were derived based on the ontology evaluation space as described in Section 3. Moreover, this model has the same nature as the ISO/IEC 25012- Data Quality model [75] which consists of only two categories namely; *inherent data quality*, and *system-dependent data quality*. However, we identified and defined three main categories for ontology quality viz: *inherent ontology quality, domain-dependent ontology quality*, and *application-dependent ontology quality* (see Table 6). The *structural intrinsic aspect* attaches to the *inherent ontology quality. The domain intrinsic and domain extrinsic aspects* are grouped under the *domain-dependent ontology quality*. Finally, the *application extrinsic* aspect was mapped with the *application-dependent ontology quality*.

In addition to the identified characteristics through the survey, a set of characteristics that can be applied to ontology were adopted from ISO/IEC 25012 namely *currentness, credibility, accessibility, availability and recoverability*

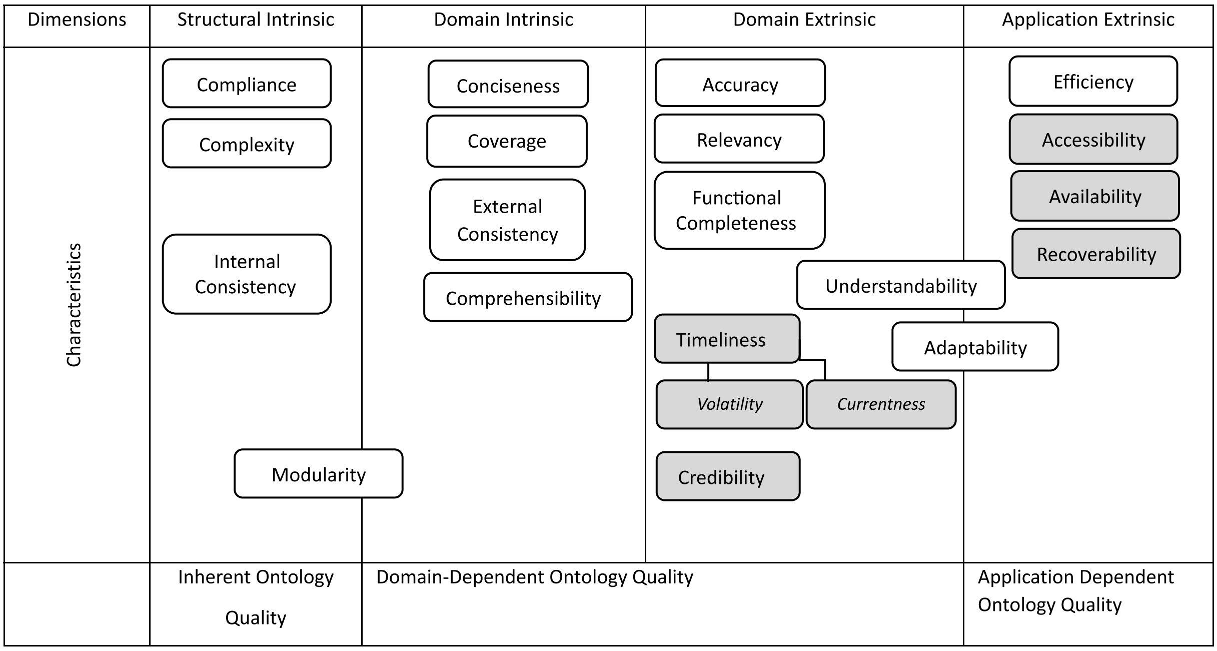
Table 5

The characteristics/attributes of the existing ontology quality models mapping with the evaluation aspects

Model	Structure Intrinsic	Domain Intrinsic	Extrinsic (Domain/Application)	Quality in use
Quality model in ONTO-EVO^{A}L [38], 2010	Complexity Cohesion Conceptualization Abstraction Comprehension	Completeness (Precision and recall)		
OntoQualitas, [114], 2014	Language conformance Completeness: *is-a/non-isa* Conciseness: *is-a/non-isa* Syntactic Correctness Semantic Correctness: *is-a/non-isa*	Completeness: *Coverage* Conciseness: *Precision* Semantic Correctness: *interpretability, clarity* Representation correctness		Usefulness
Quality model of Zhu et al. [153], 2017	Internal consistency Well-formedness Structural Complexity Modularity	External Consistency Compatibility Completeness Conciseness	Applicability Adaptability Efficiency Comprehensibility	
Quality model of McDaniel et al. [98], 2018	Lawfulness Richness Structure	Consistency Interpretability Precision Comprehensiveness	Accuracy Adaptability Ease of use Relevance	Authority History Recognition

Table 6

Ontology quality model (the gray-colored characteristics were adopted from the data quality standard: ISO/IEC 25012)



Dimensions	Structural Intrinsic	Domain Intrinsic	Domain Extrinsic	Application Extrinsic
Characteristics	Compliance Complexity Internal Consistency Modularity	Conciseness Coverage External Consistency Comprehensibility	Accuracy Relevancy Functional Completeness Understandability Timeliness Adaptability *Volatility* *Currentness* Credibility	Efficiency Accessibility Availability Recoverability
	Inherent Ontology Quality	Domain-Dependent Ontology Quality		Application Dependent Ontology Quality

[75]. Moreover, other two time-related characteristics: *timeliness* and *volatility* were identified with the characteristic: *currentness*, in which, *timeliness* depends on both *currentness* and *volatility* characteristics [152]. Altogether, twenty-one (21) characteristics were mapped with the ontology evaluation space. Each of them is explained in the following sections and definitions were derived from the discussed theories and the standard ISO/IEC 25012. Moreover, the associated measures for each characteristic have been presented in tables in the respective sections.

5.1. *Structural intrinsic characteristics (i.e., inherent ontology quality)*

The structural intrinsic aspect considers the characteristics related to the language that is used to represent knowledge and the associated inherent quality of an ontology. i.e., ontology structural properties and internal consistency. The evaluation of the characteristics that come under this aspect does not depend on the knowledge of the domain that an ontology is being modeled. This is because the evaluation of the structural intrinsic characteristics is performed based on the rules, specifications, and guidelines defined in the ontology representation language, not based on the domain knowledge that an ontology is modeled. Moreover, many characteristics have quantitative measures. As a result, many tools such as OntoQA [132], OOPS! [111], OntoMetrics [90], XD analyzer [30], ontologyAnalyzer [7,126] and Delta [86] have automated the measures of structural intrinsic characteristics. Thus, ontology developers can easily use these tools for the structural intrinsic evaluation of an ontology (see Table 13).

5.1.1. *Compliance*

In this study, the term compliance is used to denote language conformity (i.e., syntactic correctness) and adherence to the guidelines and specifications provided by the ontology language. Language conformity refers to “how the syntax of the ontology representation conforms to an ontology language” [58]. To this end, Rico et al. [114] defined the term syntactic correctness by adopting the definition provided in [24] as “the quality of the ontology according to the way it is written”. Moreover, they have used the measures of attributes: *lawfulness* and *richness* proposed by Burton-Jones et al. [24] to gauge the syntactic correctness (see Table 7). Similarly, Zhu et al. [153] stated the term: well-formedness as “syntactic correctness with respect to the rules of the language in which it is written”. Furthermore, Neuhaus et al. [102] have stated that syntactic well-formedness is an important criterion for a well-built ontology. They have discussed syntactic correctness as a part of the *craftsmanship* dimension which

Table 7

Measures of characteristics related to the structural intrinsic aspect

Characteristic	Attribute	Measures
Compliance	Lawfulness	The ratio of the total number of breached rules in the ontology is divided by the number of statements in the ontology [5,24,98,114].
	Richness	The ratio of the total syntactical features used in the ontology divided by the total number of possible features in the ontology [5,24,98,114].
	Pitfalls	The number of pitfalls related to the ontology languages (i.e., as explained in OOPS!) [111].
Complexity	Size	The number of classes, number of attributes, number of binary relationships, and number of instances [153]. The number of nodes in the ontology graph, maximal length of the path from a root node to a leaf node, number of leaves in the ontology graph; the number of nodes that have leaves among their children, and the number of arcs in the ontology graph. [50,54].
	Depth	Absolute depth, average depth, minimal depth, maximal depth; dispersion of depth; dispersion of depth divided by the average depth [50,54].
	Breadth	Average breadth; average relation of adjacent levels breadth; maximal relation of adjacent levels breadth; the ratio of dispersion of relations of adjacent levels breadth to the average relation of adjacent levels breadth [50,54].
	Fan-outness	The average number of leaf-children in a node, the maximal number of leaf-children in a node, minimal number of leaf-children in a node; dispersion of the number of leaf-children in a node [54].
	Tangledness	The number of nodes with several parents, the ratio of the number of nodes with several parents to the number of all nodes of an ontology graph; the average number of parent nodes of a node, [50,54].
	Cycles	The number of cycles in an ontology, the number of nodes that are members of any of the cycles divided by the number of all nodes of an ontology graph [50,54].
	Relationship Richness	The ratio of the number of (non-inheritance) relationships (P), divided by the total number of relationships defined in the schema (the sum of the number of inheritance relationships (H) and non-inheritance relationships (P)) [10,97,133].
		The ratio of the number of relationships that are being used by instances I_i that belong to C_i ($P(I_i, I_j)$) compared to the number of relationships that are defined for C_i at the schema level ($P(C_i, C_j)$) [133].
	Inheritance Richness	The average number of subclasses per class [133].
	Attribute Richness	The average number of attributes (slots) per class [10,133].
	Class Richness	The ratio of the number of non-empty classes (classes with instances) (C) divided by the total number of classes defined in the ontology schema (C) [133]
	Semantic Variance	Given an ontology O, which models in a taxonomic way a set of concepts C, the semantic variance of O is computed as the average of the squared semantic distance $d(\cdot, \cdot)$ between each concept $c_i \epsilon C$ in O and the taxonomic Root node of O. The mathematical expression of the semantic variance can be found in [116].
Internal Consistency	–	The number of subclass partitions with common classes, the number of subclass partitions with common instances, the number of exhaustive subclass partitions with common classes, and the number of exhaustive subclass partitions with common instances [58,114].
		The number of logical consistencies (i.e., using reasoners) [68,152].
Modularity	Cohesion	The number of ontology partitions, the number of minimally inconsistent subsets, and the average value of axiom inconsistencies [94]. The number of root nodes, maximal length of simple paths, the total number of reachable nodes from roots, the average depth of all leaf nodes [153], Average number of connected components (classes and instances) [38,133].
	Coupling	The ratio of the number of hierarchical relations that are disconnected after modularization to the total number of relations, The ratio of the number of disconnected non-hierarchical relations to the total number of relations after ontology modularization [104], the total number of relationships instances of the class have with instances of other classes [133], the number of classes in external ontologies which referenced by the discussed ontology [107].

considers "whether an ontology is well-built in a way that adheres to established best practices". In addition to that, Poveda-Villalón et al. [111] have discussed *ontology compliance*, not only by considering the syntactic correctness but also by considering the standards, i.e., specification and guidelines/styles, introduced for the ontology language (i.e., OWL [99,106]). For instance, in addition to syntax correctness, the authors in [111] have considered whether the developers use the primitives provided in the ontology implementation languages correctly. For example, if we consider Web Ontology Language (OWL), there are a few pitfalls (i.e., style issues [99,106]) related to the ontology primitives such as defining "*is*" relationship instead of using "*rdfs:subClassOf*",[3] defining a relationship itself as an inverse relationship instead of using symmetric property and defining a chain property with one relationship [111]. To detect such pitfalls, the OOPS! tool has been introduced [111]. Moreover, many ontology reasoners facilitate the detection of syntactic invalidity and style issues (i.e., guidelines) considering them as a set of ontology inconsistencies [14,152]. Accordingly, compliance can be viewed as a core feature that is a prerequisite for other quality characteristics [24,102,111]. Based on the provided definitions in works of literature, we defined compliance as in Definition 1.

Definition 1 (Compliance)**.** Compliance refers to the degree to which the ontology being constructed is in accordance with the rules, specifications and guidelines defined in the ontology representation language.

5.1.2. Complexity

Complexity describes the topological properties of an ontology [48,49]. There is no definition found in the selected papers, nevertheless, several attributes have been described such as depth, breadth, fan-outness, category size, and semantic variance, to measure the complexity of ontologies (see Table 7). Moreover, complexity is also referred to as cognitive complexity in [81,124], because it influences how well users can understand the ontological structure and to interpret knowledge that the ontology is modeled. For instance, it is difficult for humans to understand an ontology with a thousand terms [81] which thus gets high depth, breadth and tangledness [50,54]. To this end, authors in [50] have defined depth, breadth and tangledness as a set of parameters (i.e., attributes) that adversely affect ontology cognitive ergonomics. Cognitive ergonomics has been defined as a principle related to the quality of an ontology that considers whether "an ontology can be easily understood, manipulated, and exploited by end-users" [49,50]. The same principle and structural attributes (i.e., depth. breadth, fan-out, circularity) have been adopted to assess the structure of educational ontologies to observe the balance of ontology structure and its perception by users [54]. Moreover, complexity is an important characteristic that provides pieces of evidence of *redundancy*, *reliability*, and *efficiency* of an ontology [54,81,116,153]. For instance, Sánchez et al. [116] stated that "the larger the topological features (i.e., average and variance of the taxonomic depth, and the maximum and variance of the taxonomic breadth), the higher the probability that the ontology is a reliable one". On the other hand, increased complexity affects searching efficiency [41,81]. For instance, Evermann et al. [41] have shown that it would take a long time to search instances when the level of categories (i.e., concepts) is increased. Based on the facts, the complexity can be defined as in Definition 2.

Definition 2 (Complexity)**.** Complexity refers to the extent of how complicated the ontology is.

5.1.3. Internal consistency

There are different types of consistencies discussed in the previous works [14,56,58,68,71]. For instance, *structural consistency, logical consistency* and *external consistency*. *Structural consistency* refers to the syntactic validity of ontologies with respect to the language used to model the ontology [68]. This notion is considered as a part of ontology *compliance* under our study (see Section 5.1.1). *External consistency* refers to whether knowledge represented in the ontology is consistent with the relative domain knowledge [153]. Accordingly, *external consistency* evaluation is associated with domain knowledge. Thus, this notion was discussed under the domain intrinsic aspect. Under the *internal consistency*, we considered the *logical consistency* of ontologies and viewed the related definitions. Zhu et al. [153] defined internal consistency as "whether there is no self-contradiction within the ontology". The authors in [14,68] have defined internal consistency with respect to description logic as, "an ontology O is consistent if there is an interpretation I of O that satisfies every axiom in O". Hnatkowska et al. [72] in their study

[3] RDF Schema (RDFS – www.w3.org/TR/rdf-schema/.) is a specification enabling the definition of RDF vocabularies. rdfs:subClassOf: https://www.w3.org/TR/rdf-schema/#ch_subclassof.

on ontology verification have adopted the definition provided in [57,58] which states that “consistency refers to whether it is possible to obtain contradictory conclusions from valid input data”. To this end, the authors in [56,58] have explained a set of errors such as the *circularity* errors and *partition* errors (i.e., subc1ass partition with common instances, subclass partition with common classes, exhaustive subclass partition with common classes) which could lead to ontology inconsistencies. Moreover, the authors in [115] have highlighted a set of anti-patterns such as *AntiPattern AndIsOr*, *AntiPattern OnlynessIsLoneliness*, *AntiPattern UniversalExistence* and *AntiPattern EquivalenceIsDifference* which could also lead to logical inconsistencies. To this end, the authors in [18,71] have shown that the use of design patterns enables to mitigate of such anti-patterns. Poveda-Villalón et al. [111], have described the pitfalls that affect the logical consistency of an ontology and grouped them as the critical pitfalls which should be fixed. Some examples for these pitfalls are defining circles in the hierarchy, misusing “owl:allValuesFrom”, specifying the domain or the range excessively, and defining wrong equivalent/transitive/symmetric relationships. The OOPS! tool is employed to detect such pitfalls. In addition to that, reasoners can infer the logical consequences underlying knowledge representation and detect logical inconsistencies [50,54,57,58]. We adopted the definition given in [71,152] to define logical consistency for ontologies as in Definition 3.

Definition 3 (Internal consistency/Logical consistency)**.** Internal consistency refers to the extent to which the ontology is free of logical contradictions with respect to particular knowledge representation, i.e., A logical contradiction occurs when the assertion of some statement S and its denial not-S are both true.

5.1.4. Modularity

From the ontological point of view, a module is defined as “any subgraph *sg* of a graph *g*, where the set of graph elements S_1 for *sg* is such that $S_1 \subseteq S$, S is a collection of elements in the graph *g*” [50]. In the selected papers, the articles [33,104] defined a module as “a subset or a part of an ontology that is partitioned or extracted from an original ontology related to a certain topic”. Similarly, the authors in [137] have defined a module as “a subset of an ontology that captures all the knowledge the ontology contains about a given set of terms”. All the definitions mentioned so far describe what a *module* is in terms of ontology. Additionally, we found that Khan et al. [81] have provided a comprehensive definition of modularity considering the process of modularization. Mainly, the definition emphasizes five dimensions namely *use-case*, *techniques*, *type*, *annotation features* and *evaluation criteria* that are required to take into account in the modularization process. However, for this study, we considered only the definitions given for modularization from the evaluation perspective. From that perspective, in the selected papers, only two papers have given definitions for modularity. Duque-Ramos et al. [40] defined modularity as “the degree to which the ontology is composed of discrete components such that a change to one component has a minimal impact on other components”. Zhu et al. [153] stated: “how well the ontology is decomposed into smaller parts, to make it easier to understand, use, and maintain”. Additionally, other articles describe the attributes related to modularity, mainly, the cohesion and coupling attributes [88,104,107,133,153] (see Table 7). Many of the measures of these attributes are defined upon the structural notion. Some examples of the measures of modularity given from the structural notions are *the number of classes, the number of root classes, the depth of the inheritance tree*, and *the number of relationships between instances*. In addition to the structural measures, the authors in [133], proposed behavioral measures for modularity by relating the characteristics: *knowledge encapsulation* and *coverage*. Importantly, Ma et al. [94] have defined a set of ontology cohesion metrics based on the ontological semantics notion that focuses on the possible inconsistency associated with modules of the ontology such as *the number of minimally inconsistency subsets*, and *the average value of axiom inconsistencies*. Oh et al. [104] have defined the coupling measures (i.e., the number of disconnected relations of each module) that also can be used to check the consistency between modules and the original ontology. In light of these concepts, we define modularity from the evaluation perspective by adopting the definition given in [40], which is as given in Definition 4. Furthermore, as highlighted just above, a number of previous studies have assessed modularity using the structural properties of an ontology [81] (see Table 7). Nevertheless, modularity assessment may also depend on the domain knowledge as explained in [81,94]. Accordingly, modularity can be classified either as structural or domain intrinsic characteristics as shown in Table 6.

Definition 4 (Modularity)**.** Modularity refers to the degree to which the ontology is composed of discrete subsets (i.e., modules of a graph, sub-graphs) such that a change to one component has a minimal impact on the other components.

5.2. Domain intrinsic characteristics

The domain intrinsic aspect mainly considers whether an ontology for a certain application domain is modeled according to the relative domain knowledge. Thus, some domain understanding is required to assess the characteristics that come under this aspect. Consequently, the characteristics are domain-depended and evaluation can be automated with more effort while employing domain knowledge [102]. Moreover, ontology developers can use a frame of reference to make a judgment about the quality characteristics [57,58,64,150]. A frame of reference can be a standard ontology, a corpus given by experts, or a requirement specification document with a set of competency questions.

5.2.1. Conciseness

The articles [72,114,153] have adopted the definition for conciseness from [58], that is "an ontology is concise if it does not store any unnecessary or useless definitions, if explicit redundancies do not exist between definitions, and redundancies cannot be inferred using other definitions and axioms". Moreover, three types of redundancies have been explained in [58] (i) *Grammatical redundancy errors*, which occur when more than one explicit definition exists in an ontology related to the hierarchical relation either directly or indirectly. For instance, direct repetitions are: (a) defining the *is-a* relation twice between the same source and target classes, (b) defining the *instances-of* relation twice between the same instance and class. For an example of indirect repetitions, consider the three definitions explicitly defined in an ontology: A is a subclass of B, B is a subclass of C and A is a subclass of C. When considering these definitions, "A is a subclass of C" is a definition that can be inferred from the first two definitions. Thus, an explicit definition of this (i.e., A is a subclass of C) in an ontology will create an indirect repetition. Similarly, this can occur at the instances level. These types of errors (i.e., *Grammatical redundancy errors*) can be eliminated by adhering to the best practices, design patterns [6,9], and principles [65,106]) defined for ontology modeling [18,71], (ii) *Identical formal definition of some classes*: this occurs when two or more classes exist in an ontology with the same formal definition, however, with different class names. To resolve this, ontologists should identify differences between the particular classes so that they can be distinguished. Otherwise, two things can be done. One solution is to remove the duplicate definitions in order for ontology to contain only the unique formal definitions. The other solution is to define one class to be an equivalent class of the other. (iii) *Identical formal definition of some instances*: this also occurs when the same formal definition has been defined for two or more instances only differentiating them with names. To solve this error, as explained in (ii) above, ontologists should identify the different attributes of the instances in order to distinguish them. Otherwise, two things can be done. One solution is to remove the duplicate instances in a way that an ontology consists of instances only with unique formal definitions. The other solution is to define both as equivalent instances (i.e., owl:sameAs).

Rico et al. [114] have adopted the theories described under points (i), (ii) and (iii) and have defined measures to evaluate the conciseness of ontologies (see Table 8). In addition to that, conciseness has been assessed by measuring the precision of an ontology with respect to the standard ontology (i.e., a frame of reference). In this way, conciseness ensures that the ontology does not consist of any unnecessary or useless classes, relations/object properties, attributes/data properties and instances/individuals with respect to the considered domain knowledge. The definition frequently used in the previous studies was adopted in our study as in Definition 5.

Definition 5 (Conciseness)**.** Conciseness refers to the fact that all the knowledge included in the ontology is useful and precise. Thus, in an ontology, neither explicit redundancies exist between axioms nor they can be inferred using other axioms.

5.2.2. Coverage (i.e., completeness)

Completeness from a real-world (i.e., domain knowledge) perspective is considered as coverage in [10,36,107]. Authors have used the terms completeness and coverage interchangeably to describe the coverage of domain knowledge in an ontology [114]. The authors in [72,114] have adopted the definition given by Gomez-Perez [58] as "an ontology is complete if and only if: all that is supposed to be in the ontology is explicitly set out in it or can be inferred, and each definition complete". Zhu et al. [153] stated completeness is "the number of elements in the standard (i.e., a frame of reference) that are covered by the candidate ontology". As well, Ouyang et al. [107] stated that "the coverage is a number of concepts and relations with regards to the ontology set (i.e., a frame of reference)".

Table 8
Measures of characteristics related to the domain intrinsic aspect

Characteristic	Attributes	Measures
Conciseness	–	The ratio of the number of classes with the same formal definition as other classes in the ontology divided by the number of classes in ontology, the ratio of the number of instances with the same formal definition as other instances in the ontology divided by the number of instances in the ontology, The ratio of the number of redundant subclass-of relations in the ontology divide by the number of hierarchical relations, The ratio of the number of redundant non-hierarchical relations in the ontology divided by the number of non-hierarchical relations, The ratio of the number of redundant instance-of relations in the ontology divided by the number of instance-of relations in the ontology [114].
	Precision	The ratio of the number of classes matches between the candidate ontology and the classes in a frame of reference (i.e., standard ontology) divided by the number of classes in the candidate ontology (this measure can be extended for other entities instead of classes such as relations, features, instances) [114].
Coverage	Recall	The ratio of the number of matching entities (i.e., class, relations, instances, terms in case of data extracted from a corpus) between candidate ontology and the frame of reference (i.e., a standard ontology) divided by the total number of entities in the standard ontology [36,114,153].
	Precision	The ratio of the number of classes matches between the candidate ontology and the classes in a frame of reference (i.e., standard ontology) divided by the number of classes in the candidate ontology (this measure can be extended for other entities instead of classes such as relations, features, instances) [36,130].
	F-measure	The harmonic means between the Recall and Precision metrics [36].
External Consistency	–	Whether ontology users disagree on the validity of the (potential) instances of the ontology elements [2].
	Interpretability	The ratio of the number of terms that have a sense listed in an independent authority divided by the total number of terms used to define classes and properties in the ontology [5,98,114].
	Clarity/Precision	The ratio of the total number of terms used to define classes and properties in the ontology divided by the number of definitions for terms in an independent authority that occur in the ontology [5,98].
		The ratio of the number of axioms (i.e., class axioms, property axioms, instance axioms, etc.) overlaps with the frame of reference divided by the total number of axioms defined in the ontology [150].
	Semantic Richness	The ratio of correct concepts, the average ratio of correct instances, the average ratio of correct attributes, Average ratio of correct relations [153].
Comprehensibility	–	The ratio of annotated classes is divided by the total number of classes, the ratio of annotated instances is divided by the total number of instances, the ratio of annotated semantic relations (object properties) is divided by the total number of semantic relations [10,38,50,98].
	Vagueness	The ratio of the number of ontological elements (classes, relations, and data types) that are vague divided by the total number of elements, the ratio of the number of vague ontological elements that are explicitly identified divided by the total number of vague elements [36].
	Clarity	The average number of word senses per unique word, and divide that value by the total number of unique words [5], the number of word senses for the term/s (i.e., classes or properties) in WordNet [24].
	Interpretability	The ratio of the number of terms that have a sense listed in an independent authority divided by the total number of terms used to define classes and properties in the ontology [5,98,114].

Thus, in the domain intrinsic aspect, completeness can be considered as the coverage of structure, content and design (i.e., concepts, instances, relations, and constraints) that can be determined concerning the domain knowledge being modeled. Accordingly, the definition was derived for completeness as in Definition 6. Moreover, incompleteness may occur due to missing disjointness, missing domains and ranges, missing necessary and sufficient conditions, and missing existential & universal restrictions [58,111,114]. These are some pitfalls that can be detected under the characteristic: *compliance*. Thus, compliance is a prerequisite to completeness in this domain intrinsic aspect.

Definition 6 (Coverage)**.** Coverage refers to the degree to which an ontology covers the axioms which have been specified (i.e., requirement specifications, standard ontologies, standard corpus) with respect to the domain knowledge that the ontology was developed to represent.

5.2.3. *External consistency*

Zhu et al. [153] defined external consistency as "whether the ontology is consistent with the subject domain knowledge". Vrandečić [142] stated that external consistency can also be named as domain coherence. Hnatkowska [72] and Rico et al. [114] adopted the consistency definitions provided in [58]. The author in [58] has also discussed the different notions of consistency as we highlighted in Section 5.1.3. With respect to external consistency, the author has defined ontology consistency as "the interpretation of definitions (formal/informal) should be consistent with respect to the real-world" [58]. For instance, if the term *Monday* is defined as a *month* in an ontology, then, that definition is inconsistent in relation to the domain knowledge. Moreover, the author in [58] has discussed these types of inconsistencies as *incorrect semantics*. A few more examples are (i) *cat* class is defined as a subclass of *house*, (ii) *tom* who is a *cat* in the real-world defined as an instance-of *house*, (iii) the relationship *eats* is defined between *cat* and *house*. The use of development methodologies, and adhering to design patterns/principles would prevent such modeling errors and mistakes (i.e., as we highlighted in Section 3).

Neuhaus et al. [102] defined fidelity as "whether the ontology represents the domain correctly, both in the axioms and in the annotations that document the ontology for humans". This definition is also similar to the definitions provided for external consistency in [153] and semantic correctness in [34,57,58]. Thus, all definitions are referred to the same characteristic and can only be evaluated relative to the subject domain. Thus, it is required to define the frame of reference to assess the ontology [56,58,150]. For instance, if an ontology developer needs to ensure that the *class axioms* defined in an ontology are consistent with the domain knowledge, then it is possible to examine the *class precision* (see Table 8). *Class precision* can be calculated by dividing *the number of classes matches between the candidate ontology and the frame of reference* by *the total number of classes in ontology*. Based on the provided definitions in the works of literature, we derived external consistency/semantic accuracy as in Definition 7.

Definition 7 (External consistency/Semantic correctness)**.** External consistency refers to the degree to which an ontology (i.e., ontology axioms) is coherent with the specified domain knowledge (i.e., requirement specifications, standard ontologies, standard corpus) that the ontology was developed to represent.

5.2.4. *Comprehensibility*

The authors of the articles [10,38,102] have described comprehensibility as the level of annotations that facilitate understanding the ontology. To assess comprehensibility, the measures: *the average number of annotated classes, the average number of annotated relations*, and *the average number of annotated instances per class*, have been used (see Table 8). In the linked data quality assessment, Zaveri et al. [152] show that comprehensibility has been interchangeably used with understandability. Same for ontologies, Poveda-Villalón et al. [111] and McDaniel et al. [98] evaluate ontology comprehensibility in terms of understandability using the same measures similar to [10] such as the *number of annotations per term in the ontology, the number of missing and misusing annotation*s. Basically, these are usability-related measures (i.e., level of annotations) as described in [48–50] which are related to the communication context of an ontology. Also, it has been named as "*ease of use"* in [98]. To this end, Poveda-Villalón et al. [111] have revealed a set of pitfalls related to the ontology usability-profiling (i.e., ontology understandability and clarity) such as *missing annotations* (i.e., metadata), *misuse of annotations*, and *use of different naming criteria*. The authors [111] state that correcting these pitfalls is required to enable users (i.e., ontology developers and consumers) to easily recognize and understand the ontology elements. Moreover, the proper use of annotations (i.e., metadata) and the same naming criteria in ontology modeling have been identified as best practices that ensure the comprehensibility of ontologies enabling clean and consistent knowledge representations [9,49]. To this end, the presentation ontology design patterns (ODPs) have been introduced in [9] that consist of two ODPs namely Annotation ODPs and Naming ODPs. Furthermore, as highlighted under Section 5.1.2, the increased complexity makes an ontology difficult to comprehend and ontology modularization is one of the solutions for it [81,153]. Based on these facts, we derived comprehensibility as in Definition 8.

Definition 8 (Comprehensibility)**.** Comprehensibility refers to the degree of annotations (i.e., metadata) of an ontology and how its elements enable users (i.e., ontology developers and consumers) to understand the appropriateness of the ontology for a specified context of use.

5.3. Domain extrinsic characteristics

From the *domain extrinsic* point of view, quality evaluation is performed taking into account an ontology as a whole without peering into the internal structure and design. It considers whether the ontology meets the domain requirements that are specifically needed for the particular use case (i.e., the context of use) which is also defined as the fitness of the ontology in [102]. To this end, Neuhaus et al. [102] stated that "successful answers to competency questions provide evidence that the ontology meets the model requirements that derive from query-answering based functionalities of the ontology". Moreover, the characteristics associated with this aspect are functional and subjective as the evaluation is performed with respect to the specified tasks and domain requirements of users (i.e., ontology consumers' views, domain experts, application users, agents of intelligent systems). Hence, the support of domain experts and users is required for evaluating the characteristics that come under this aspect.

5.3.1. Accuracy (i.e., functional correctness)

The selected papers have not provided definitions for accuracy from the domain extrinsic aspect. However, the authors in [5,98] have adopted measures from the semiotic metric suit of Burton-Jones et al. [24]. According to them, accuracy is "whether the claims an ontology makes are true". Similar to this definition, Duque-Ramos et al. [40] defined *precision* under functional adequacy as "the degree to which the ontology provides the right or specified results with the needed degree of accuracy". The authors in [130] have used answers provided to the competency questions to assess functional correctness (see Table 9). ISO/IEC 25010 provides a definition for functional correctness as "the degree to which a product or system provides the correct results with the needed degree of precision" [77]. This definition can be adopted for ontology under the domain extrinsic aspect because the ontology is considered as a whole and a part of an information system under the domain extrinsic aspect [102]. Accordingly, we defined the accuracy of ontologies as in Definition 9.

Definition 9 (Accuracy/Functional correctness)**.** Accuracy refers to the degree to which an ontology provides the correct results (i.e., information and knowledge) with the needed degree of precision.

5.3.2. Relevancy

From the selected articles, only Amith et al. [5] have provided a definition for relevance as "fulfillment of a specific use case". McDaniel et al. [98] adopted measures for relevance from [24], in which relevancy is defined as "whether the ontology satisfies the agent's specific requirements". The ISO/IEC 25010 standard defines *functional appropriateness* which is similar to the definition of relevance "the degree to which the functions facilitate the accomplishment of specified tasks and objectives". From the ontological point of view, we define relevancy as in Definition 10. It can be measured by assessing the percentage that *the ontology appropriately answers the competency questions specified in the context of use* [5,130]. For instance, with respect to the use case in agriculture, consider a scenario where a farmer queries the control methods for the late blight disease in potatoes. According to the present agriculture domain, one control method for late blight is "applying fungicides: mancozeb at 0.25%

Table 9

Measures of characteristics related to the domain extrinsic aspects

Characteristic	Measures
Accuracy	The ratio of the number of false/true statements in the ontology is divided by the number of statements in the ontology [5,24,98]. The number of competency questions that are correctly answered [130].
Relevance	The ratio of the type of syntax relevant to the user is divided by the number of statements in the ontology [24,98], the percentage of adherence to the competency questions [5].
Functional Completeness	There is no significant measure that has been defined for completeness in the functional aspect. Thus, based on the definitions, the measure "the number of competency questions that have provided sufficient answers concerning the context of use" was derived.
Understandability	There is no significant measure that has been defined for understandability in the functional aspect. Based on the definitions, the measure was derived as follows: "The number of competency questions that received answers in the human-readable language (or the language that can be interpreted to other languages) with appropriate context terms, symbols and units in a specified context is divided by the total number of competency questions.

followed by cymoxanil+mancozeb or dimethomorph+mancozeb at 0.3% at the onset of disease". However, the fungicide percentage to be considered may vary from one farm location to another and it may also depend on the farm size. If all farmers are provided with the aforementioned control method for late blight in potatoes by the ontology, some farmers who are in locations with different conditions than the contexts normally considered may not find these solutions to be applicable. In this situation, it can be observed how many competency questions are answered relevant to the considered context by an ontology in order to measure the relevancy of an ontology.

Definition 10 (Relevancy)**.** Relevancy refers to the degree to which an ontology provides appropriate information and knowledge with respect to the specified context of use.

5.3.3. Functional completeness

Only Poveda-Villalón et al. [111] have discussed the completeness from the functional aspect/end-user perspective in the selected articles. They defined completeness as "the coverage of the requirements specified in the ontology requirement specification documents by the ontology". Fox and Grüninger [44] defined functional completeness as "the representation of necessary information by an ontology for a function to perform its task". Furthermore, they stated that "the functional completeness of an ontology is determined by its competency". For that, the success of answering the competency questions can be analyzed related to a particular function (i.e., use case). Similarly, in data quality, completeness is defined as "the extent to which data are of sufficient breadth, depth, and scope for the task at hand" [146]. For software quality, The ISO/IEC 25010 standard defines functional completeness as "the degree to which the set of functions covers all the specified tasks and user objectives". Based on these facts, we derived functional completeness of ontologies as in Definition 11.

Definition 11 (Functional completeness)**.** Functional completeness refers to the degree to which an ontology provides necessary and sufficient information and knowledge with respect to the specified context of use.

5.3.4. Understandability

From the ontological point of view, many authors such as in [10,38,49,98,101,102,111] have discussed understandability as an ontology intrinsic characteristic. Mainly, understandability has been assessed by observing internal attributes of ontologies such as *annotations* and *naming conventions*. We discussed this notion of understandability under the domain intrinsic aspect by defining the term *comprehensibility* (see Section 5.2.4). Nonetheless, understandability can also be viewed as a domain extrinsic characteristic in terms of how easily users (i.e., ontology consumers, end-users) can understand the knowledge (i.e., answers to competency questions) provided through the ontology. To this end, Zhu et al. [153] defined the understandability of an ontology as "whether human readers can easily understand the semantic description" given in the ontology. The data quality standard: ISO/IEC 25012 also describes understandability from the extrinsic aspect. It defines understandability as "the degree to which data has attributes that enable it to be read and interpreted by users, and are expressed in appropriate languages, symbols and units in a specific context of use" [75]. In addition to that, understandability can be viewed as a characteristic associated with the application extrinsic aspect when an application interprets the knowledge retrieved from the ontology. For instance, in an ontology-based IS, end-users access ontologies through software applications rather than directly accessing the ontology. In this situation, information and knowledge provided through the ontology are required to be translated by the software application. Thus, from the application extrinsic aspect, understandability can be defined as how easily an application can interpret the knowledge that is retrieved from ontologies in order to express them in appropriate languages/formats that the end-users/agents can understand. Consequently, understandability has been mapped by associating both domain and application extrinsic aspects in Table 6. Moreover, we defined understandability from the extrinsic aspect as in Definition 12 by adopting the definition given in the data quality standard: ISO/IEC 25012.

Definition 12 (Understandability)**.** Understandability refers to the degree to which the information and knowledge provided through the ontology can be comprehended, without ambiguity and is expressed in appropriate languages, symbols and units in a specific context of use.

5.4. Application extrinsic characteristics

Under the application extrinsic aspect, ontology quality assessment is performed by considering an ontology as a component of a system. It considers the requirements of an ontology that are needed by a particular application in that the ontology is integrated. Thus, application extrinsic quality attaches to the capabilities of reasoning tools, computer systems' components: hardware/software, the technical environment in which the ontology is used, and the tasks to be performed. The characteristics relevant to this aspect are evaluated from the application perspective and this evaluation is independent of domain knowledge. Ontology developers can assess the quality characteristics which are relevant to this aspect with the support of domain experts, software quality engineers and, perhaps with the support of domain users.

5.4.1. Adaptability

Adaptability has been frequently discussed under ontology evolution. Ontology evolution has been defined as "timely adaptation of an ontology with respect to certain requirements by maintaining the consistency" [43,67,68]. Similarly, the authors in [58,153] have defined adaptability as how easily ontology can be changed with certain requirements. The requirements for ontology changes may occur due to (i) changes in user needs, (ii) changes in the application needs that the ontology is integrated with, and (iii) changes in ontology conceptualization [43]. These changes could lead to adding, removing or modifying axioms in the ontology. Sometimes, when changes are performed to the ontology, these may cause inconsistencies (i.e., structural, logical, domain/external). Thus, to avoid such inconsistencies, some other additional changes to the ontology are required to be performed. To this end, the authors in [56,58] stated that the changes should be performed without altering the axioms already guaranteed. In the selected survey papers, Zhu et al. [153] defined the adaptability of an ontology in the context of web services as "how easily the ontology can be changed to meet the specific purposes of developing a particular web service". Moreover, the sub-factors such as *tailorability*, *composability, extendability*, and *transformability* have been defined under adaptability. McDaniel et al. [98] adopted the adaptability definition from Vrandečić [142] which is "adaptability measures how well an ontology anticipates, how its future uses and whether it provides a secure foundation which is easily extended and flexible enough to react predictably to small internal changes". Originally, this definition has been provided by Gomez-Perez [58] referring to *expandability* and *sensitiveness*. In addition to that, Gangemi et al. [49,50] defined adaptability in terms of flexibility as "an ontology that can be easily adapted to multiple views". In which *modularity* and *partition* are defined as the attributes related to adaptability. Based on the provided definitions, we derived adaptability as in Definition 13. When considering the evaluation of adaptability, the authors in [49,50,98] have assessed adaptability using quantitative measures which are independent of domain knowledge. However, the authors in [43] have highlighted that adaptability can be assessed by analyzing the answers given to competency questions after modifying the ontology with respect to the changes (see Table 10). Moreover, ontology changes may occur due to the domain and/or application requirements as pointed in (i) and (ii). Therefore, adaptability can be considered as a characteristic that can come under both domain and application extrinsic aspects (see Table 6).

Table 10

Measures of characteristics related to the application extrinsic aspects

Characteristic	Attributes	Measures
Adaptability	–	The sum of the average number of ancestors for the leaves in an ontology and the ratio of the number of leaves to the total number of classes in an ontology [98]. The number of competency questions was correctly answered after the changes [43].
Efficiency	Size	The sum of the average number of classes, the average number of sub-class per class, the average number of relations, the average number of relations per class, and the average ontology size [10].
	–	Response time, the number of resources utilized [40]. disjointness ratio [50], The number of cycles in an ontology, the number of nodes that are members of any of the cycles is divided by the number of all nodes of an ontology graph [50,54], Tangledness: number of nodes with several parents, the ratio of a number of nodes with several parents to a number of all nodes of an ontology graph; the average number of parent nodes of a graph [50].

Definition 13 (Adaptability)**.** Adaptability refers to the effort required to change (i.e., add, remove, modify) the ontology definitions (i.e., axioms) without altering the definitions that are already guaranteed.

5.4.2. *Efficiency*

In this study, the term efficiency refers to ontology computational or performance efficiency. The articles [8,10,91] adopted the definition of computational efficiency given by Gangemi et al. [48–50] as "an ontology that can be successfully/easily processed by a reasoner". On the other way, it is the response time and memory consumption utilized by reasoners when answering queries, classification, or checking consistency. Gangemi et al. [49,50] proposed measures related to computational efficiency. They are disjointness ratio, tangledness, circularity, and restrictions. Additionally, Evermann et al. [41] provided empirical evidence to prove that semantic distance and category size (i.e., the number of instances) influence to search efficiency of an ontology. Similarly, Bouiadjra et al. [10] claimed that the measures of the size such as *the average number of classes, the average number of sub-classes per class, the average number of relations, and the average number of relations per class*, can be adapted to assess the efficiency. Duque-Ramos et al. [40] defined two sub-characteristics of performance efficiency, (i) *Response time*: "the degree to which the ontology provides appropriate response and processing times and throughput rates when performing its function, under stated conditions". (ii) *Resource Utilization*: "the degree to which the application uses appropriate amounts and types of resources when the ontology performs its function under stated conditions". Based on these facts, the definition for efficiency of an ontology was derived as in Definition 14.

Definition 14 (Efficiency)**.** Efficiency refers to the degree to which the ontology can be processed and provide the expected level of performance by utilizing the appropriate amount and types of resources in a specific context of use.

5.5. *Time-related characteristics adopted from ISO/IEC 25012*

Batini et al. [13] proposed three types of time-related dimensions such as currentness, timeliness, and volatility which are interchangeably discussed in the literature. Zaveri et al. [152] have adopted these characteristics for linked data quality assessments and have shown timeliness depends on characteristics: currentness and volatility. Each of these characteristics was grouped under the dataset dynamicity aspect. Similarly, these three characteristics would become applicable for ontologies as the knowledge that is considered for the representation can constantly change and expand in the real-world use case [43,67,133]. For instance, in the medical domain, constantly new diseases and respective treatments can be discovered. If an ontology-based application is developed to share such knowledge, then the ontology integrated with that application should be updated once the new disease knowledge is discovered and it should be made available to the respective users. Thus, we discussed the three time-related characteristics concerning the dynamic nature of an ontology, and possible measures are presented in Table 11.

5.5.1. *Currentness*

For data quality, currentness is defined in ISO/IEC 25012 as "the degree to which data has attributes that are of the right age in a specific context of use". Batini et al. [13] defined currentness as "how promptly data are updated". For linked data, Zaveri et al. [152] adopted the same definition given by Batini et al. [13]. There is no definition specifically defined for ontology. However, ontology evolution is required over time while domain information is changed as mentioned in Section 5.4.1. We adopted the definition provided by Batini et al. [13] for ontologies as in Definition 15.

Definition 15 (Currentness)**.** Currentness refers to how promptly the ontology knowledge is updated.

5.5.2. *Volatility*

Batini et al. [13] stated volatility is "the frequency with which data vary in time". Similarly, for ontologies, Murdock et al. [100] defined volatility as "a measure of the amount of change between two or more different versions of a populated ontology". Stvilia [124] defined it as "the amount of time the content of an ontology remains valid" and can be measured by calculating the average update rate of the ontology. Volatility is a property that assesses the stability of an ontology. Information such as dates of birth, places of birth, and manufacturing dates has zero volatility. Information such as stock exchange prices have a high degree of volatility, thus it is valid for a very short time. Based on these facts, we derived the volatility of ontologies as in Definition 16.

Definition 16 (Volatility)**.** Volatility refers to the frequency of which the content of the ontology remains valid.

Table 11
Measures of characteristics adopted from ISO/IEC 25012

Characteristic	Attributes	Measures
Timeliness	–	Difference between the last modified time of the original/target sources and the last modified time of the particular knowledge in the ontology [46,152].
Currentness	–	Average class currency [124] (This measure can also be extended for other entities of an ontology) Average update rate [124].
Volatility	–	The amount of change between two or more different versions of a populated ontology [100]. Average update rate [124].
Credibility	Authority	The number of other ontologies that link to the target ontology, and the number of shared terms there are within those linked ontologies [97].
	History	The number of times ontology has been used, the number of years the ontology is in the open resource, and the number of revisions made [97]. The number of positive user feedback given on the ontology (i.e, information and knowledge).
Accessibility	–	The measures of characteristics: complexity and modularly (see Table 7), comprehensibility (see Table 8), and semantic accuracy (see Table 8) can be adopted [50].
Availability	–	This has been measured w.r.t. YES/NO scenarios. Thus, the given questions can be used to check availability. “is an ontology made available to the application that the ontology is integrated (i.e., as RDF file/as HTML)? [111], is documentation of an ontology made available? [111], is an ontology consisting of URI without any supporting RDF metadata? [152], is an ontology consisting of dead links?” [152].
Recoverability	–	There are no specified measures that have been provided. This can be qualitatively evaluated by considering an ontology as a software artifact [40].

5.5.3. Timeliness

It is not only enough ontology is updated, but also the information and knowledge of an ontology should be made available timely for a specified task [13]. For example, assume a timetable of a particular course unit has been updated. Although it already contains updated data, it may not be timely and will become useless if the timetable can only be made available for students after starting the classes. From this aspect, there is no definition found in the literature on ontologies. Thus, we adopted the definition given for data quality by Batini et al. [13] for ontologies as in Definition 17.

Definition 17 (Timeliness)**.** Timeliness refers to how up-to-date information and knowledge of an ontology are for a specified task.

5.6. Other characteristics adopted from ISO/IEC 25012

5.6.1. Credibility

Credibility is the quality of being trusted and believed in [28]. For data quality, credibility is defined as “the degree to which data has attributes that are regarded as true and believable by users in a specific context of use” [75]. The term believability has been used interchangeably to describe credibility. Wang et al. [146] defined believability as “the extent to which data are accepted or regarded as true, real, and credible”. The product acceptance by the community depends on how it can be trusted. From an ontological point of view, McDaniel et al. [97] have discussed ontology quality attributes such as *authority* and *history* that determine community acceptance of ontologies. These are also attributes of the social quality of ontologies that initially were defined in the semiotic metric suite by answering the question “Can ontology be trusted” [24]. Moreover, the semiotic metric suite has defined *authority* as “the extent to which other ontologies/ ontology consumers rely on it”. *History* is another attribute that is evaluated upon: *the number of times that an ontology has been used, the number of years ontology in the public library*, and *the number of revisions made*. By adopting the definition provided in ISO/IEC 25012 [75], we derived the definition for credibility from the ontological point of view as in Definition 18.

Definition 18 (Credibility)**.** Credibility refers to the extent to which an ontology is accepted or the information and knowledge provided through ontologies regards as true and believable with respect to the specified context of use.

5.6.2. Accessibility

It is important to consider how easy an ontology can be accessed for the specified purpose. The articles that we have selected for this survey have not provided a definition for accessibility. In a milestone paper, Gangemi et al. [49, 50] defined accessibility as "an ontology that can be easily accessed for effective application". Vrandečić [142] has adopted the same definition provided by Gangemi et al. [49,50]. The supported sub-characteristics for accessibility, as defined by Gangemi et al. [49,50], are *modularity, logical complexity, annotation*, and *accuracy*. Additionally, the authors have stated that these sub-characteristics positively influence accessibility, except logical complexity. However, they have not provided a clear explanation of how those sub-characteristics are positively/negatively associated with accessibility. For example, if we consider *annotation*, we can realize that providing proper meta-information about the *ontology content, its components, and configuration* helps ontology users to understand the ontology. Thus, it makes easy for ontology users to decide which knowledge to be accessed. In such a way, it should be explained how the aforementioned sub-characteristics affect accessibility, however, it has not been provided in [49,50].

From the data quality perspective, ISO/IEC 25012 defines accessibility as "the degree to which data can be accessed in a specific context of use, particularly by people who need supporting technology or special configuration because of some disability". Based on the definition given in [49,50] for accessibility from the ontological perspective and based on the ISO/IEC 25012 definition, we derived the definition for accessibility as in Definition 19.

Definition 19 (Accessibility)**.** Accessibility refers to the extent to which an ontology can be easily accessed.

5.6.3. Availability

Duque-Ramos et al. [40] defined availability for ontologies by assuming ontology as a software artifact in an application, which is "the degree to which a software component (language, tools, and the ontology) is operational and available when required for use". Poveda-Villalón et al. [111] stated that "ontology not availability" is a critical pitfall in a situation where a system entirely depends on the ontology. We adopted the definition given in [152] for ontology availability as in Definition 20.

Definition 20 (Availability)**.** Availability refers to the extent to which the knowledge represented in the ontology (or some portion of it) is present, obtainable and ready for use.

5.6.4. Recoverability

Duque-Ramos et al. [40] defined recoverability as "the degree to which the ontology can re-establish a specified level of performance and recover the data directly affected in the case of a failure". This definition describes the extent to which an ontology can regain its *performance level* after experiencing some failure. However, when considering the definition given in ISO/IEC 25012, *recoverability* is described in a broader aspect, which is "the degree to which data has attributes that enable it to maintain and preserve a specified level of operations and quality, even in the event of failure, in a specific context of use". None of the other selected papers have discussed recoverability. Recoverability is an important characteristic to be considered as ontology quality can fail for multiple reasons. For instance, invalid modifications and inappropriate changes made to an ontology can lead to quality issues [85,133]. Thus, keeping different versions of an ontology is useful to track changes, detect invalid modifications, detect inconsistencies, and to re-establish the specified level of quality in the case of failure [67,133]. Accordingly, the definition for recoverability was derived as given in Definition 21 by adopting the definition provided in ISO/IEC 25012.

Definition 21 (Recoverability)**.** Recoverability is the degree to which the ontology maintains and preserves a specified level of quality, even in the event of failure with respect to a specific context of use.

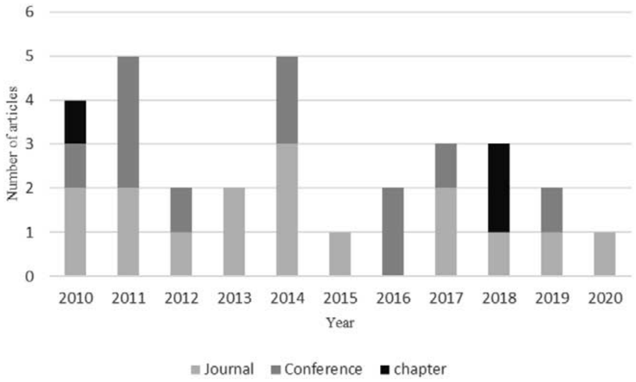


Fig. 6. Distribution of the core survey papers across the years.

6. Discussion

In our survey, we selected 30 core papers for review and these papers consisted of sixteen journals, eleven conference papers, and three chapters. The distribution of articles across the years is presented in Fig. 6.

In Table 12, the number of characteristics considered in each article was mapped along with the characteristics defined under the evaluation aspects in Table 6. The authors in [102] have not specifically discussed a set of characteristics. However, they have provided five high-level characteristics covering a set of questions attached to the ontology quality assessment. For instance, they [102] discuss high-level characteristics namely *intelligibility* (i.e., can humans understand the ontology correctly?), *fidelity* (i.e., does the ontology accurately represent its domain?), *craftsmanship* (i.e., is the ontology well-built and are design decisions followed consistently?), *fitness* (i.e., does the representation of the domain fit the requirements for its intended use?), and *deployability* (i.e., does the deployed ontology meet the requirements of the information system of which it is part?). Under each characteristic, a concise description of the quality to be focused on has been provided. The article [8] has evaluated ontology named U ontology (i.e., representation of the ontology usage analysis domain) by adopting eight characteristics proposed in [142]. Nevertheless, it is unclear how the considered characteristics (i.e., adaptability, conciseness, and fitness) were evaluated.

In summary, the least discussed characteristics in the selected studies are *credibility, volatility, efficiency*, *accessibility, availability*, and *recoverability* (see Table 12). None of the selected studies has considered *timeliness* and *currentness* for ontology quality assessments. However, there were frequent discussions on the characteristics compared to others in descending order: *complexity, coverage, external consistency, internal consistency, modularity, compliance*, and *comprehensibility* (see Table 12). Most of them come under the structural intrinsic aspect. Accordingly, it can be observed that the existing studies have provided less attention to the extrinsic aspects of the ontology, particularly after the ontology is deployed in a system. This issue has further been realized when analyzing the evaluation tools and methods (see Section 7.4). From that analysis, it can be shown that there are many tools available for evaluating intrinsic ontology quality, but only a few tools or methods are available for evaluating extrinsic quality.

When considering the terms and definitions of ontology quality, none of the works has made a clear distinction between the concepts of quality characteristics, attributes and measures from an ontology quality point of view. Because of this issue, ontologists have used these terms interchangeably. For instance, although *clarity* has been defined as an evaluation *characteristic/criterion* in [62], in [24,114], it has been defined as an *attribute*. Additionally, it can be seen that various definitions for one quality characteristic have been provided in the literature (see Section 5). This can create confusion among ontology developers and researchers, as they may not know which definition to use or which one is the most appropriate for their situation.

Another significant issue in the existing studies is that, except for the research [114,153], all other studies have used a common set of characteristics without taking the *context of use* and *the user needs* into account when defining ontology quality characteristics. As highlighted in Section 1, quality is the fitness of a product for the intended

Table 12
The respective characteristics discussed in each of the selected approaches with the evaluation aspects

Approaches/Characteristics	Compliance	Complexity	In. Consistency	Modularity	Conciseness	Coverage	Ex. Consistency	Comprehensibility	Accuracy	Relevancy	Functional Completeness	Timeliness	Volatility	Currentness	Credibility	Understandability	Adaptability	Efficiency	Accessibility	Availability	Recoverability
Evermann et al. [41], 2010		x																			
Ma et al. [94], 2010				x																	
Djedidi et al. [38], 2010		x		x		x		x													
Tartir et al. [133], 2010		x		x																	
Oh et al. [104], 2011				x																	
Duque-Ramos et al. [40], 2011	x	x	x	x	x	x	x		x	x	x						x	x	x	x	x
Murdock et al. [100], 2011							x						x								
Bouiadjra et al. [10], 2011		x				x		x										x			
Ouyang et al. [107], 2011				x		x															
Gavrilova et al. [54], 2012		x																			
Schober et al. [118], 2012								x													
Haghighi et al. [34], 2013			x		x	x	x				x						x				
Neuhaus et al. [102], 2013	x						x	x	x	x	x						x				
Sánchez et al. [116], 2014		x																			
Poveda-Villalón et al. [111], 2014	x		x			x	x	x												x	
Alexopoulos et al. [2], 2014							x														
Rico et al. [114], 2014	x		x		x	x	x														
Batet et al. [12], 2014		x																			
Lantow [91], 2016		x		x																	
McDaniel et al. [97], 2016															x						
Hnatkowska et al. [72], 2017			x		x	x															
Zhu et al. [153], 2017	x	x	x	x	x	x	x									x	x	x			
Tan et al. [130], 2017	x		x						x	x											
Ashraf et al. [8], 2018		x	x			x					x										
McDaniel et al. [98], 2018	x	x				x	x	x	x	x					x		x				
Kumar et al. [88], 2018		x		x		x															
Demaidi et al. [36], 2019						x															
Amith et al. [5], 2019	x	x					x		x	x					x						
Franco et al. [45], 2020		x																			

use (i.e., requirements/needs) [31]. From the ontological point of view, ontology quality is the fitness of the representation of the domain knowledge for its intended use [102,114]. Therefore, the quality characteristics which are to be achieved by ontologies should be specified with respect to the *intended needs* in the specified *context of use*. Based on this fact, it can be stated that all the characteristics presented in Table 6 are not equally important for all contexts of use. However, the characteristics such as *compliance, internal consistency, external consistency, coverage, accuracy, functional completeness* and *timeliness* can be viewed as core characteristics that are required for ontologies regardless of the context of use. In addition to these, there are a set of characteristics that are necessary to achieve through an ontology with respect to the intended needs and those characteristics should be properly identified. For instance, when considering the ontology-based decision support system in agriculture, relevancy is a crucial factor as stated in [145]. Moreover, the authors in [81] have exemplified that the modularity characteristic is not applicable to all ontologies and it depends on the needs of users and applications. In addition to that, authors in [114,150] have stated that conciseness of ontologies is not a required characteristic when an ontology is modeled for extensive coverage of the domain. Accordingly, it can be understood that the necessary and sufficient quality characteristics for an ontology should be identified in relation to the *needs* of the considered use case (i.e., the *context of use*) [114,150]. Then, these identified characteristics should be assessed across the ontology development life cycle [102].

7. Important gaps identified through the survey

In light of our survey, the important research gaps further to be investigated have been identified such as (i) the absence of relational quality models for ontologies, (ii) the lack of systematic approaches for ontology quality specification and evaluation (iii) the lack of detailed guidelines for selecting the correct modeling choices at the conceptual level (iv) the lack of methods and tools for assessing the extrinsic quality. The subsequent sections further explain these important gaps with the possible solutions.

7.1. Relational quality models for ontologies

It is vital to explore dependencies between the characteristics as one characteristic may affect another characteristic either negatively or positively. Up to now, there are no relational (i.e., non-hierarchical) models that show the correlations among ontology quality characteristics. However, significant discussions on certain characteristics and attributes have been made in the literature. Evermann et al. [41] stated that the searching time is increased when the category size (i.e., the number of instances per class) and semantic distance are high. Moreover, the authors in [12,116] have used complexity attributes: *semantic variance of the taxonomies depth and breadth* for evaluating *semantic accuracy* as they are positively correlated, which was verified in [42]. Franco et al. [45] have analyzed the correlation between structural measures. This is useful for researchers to reduce the number of measures to be examined in the quality evaluation by omitting the measures that show the same impact. Additionally, in the context of semantic descriptions of web services, Zhu et al. [153] made possible assumptions such as; (i) conciseness, structural complexity, and modularity affect adaptability; (ii) conciseness would reduce the complexity of service descriptions; (iii) efficiency may be reduced if the structural complexity is higher. Similarly, Sánchez et al. [116] claimed that reliability is relatively high when an ontology consists of more topological features. However, valid empirical evidence is required to confirm such correlations. For instance, it has been declared that tangledness negatively affects computational efficiency in [50], however, Yu et al. [149] empirically proved that ontology tangledness positively impacts efficiency when an ontology is specified for browsing articles. Thus, it is required more empirical evaluations to confirm the correlation between the characteristics instead of superficial investigations [142]. Furthermore, empirically validated relational models for ontologies are essential that can be used as a foundation for future research rather than performing experiments from the beginning and over again.

7.2. Lack of systematic approaches for ontology quality specification and evaluation

Although quality evaluation is considered as a phase of ontology development methodologies [64,120,128,139], it is an iterative process that should be started from the requirement analysis and specification phase. For example, quality assessment consists of several activities such as; the *identification of intended needs through stakeholder discussions* (*i.e.*, *requirement specification*), *elicitation of quality requirements from the identified needs* (*i.e.*, *quality specification*), *prioritizing the quality requirements*, *specifying quality characteristics* (*i.e.*, *intrinsic and extrinsic aspect*) *and performing quality assessment across the development* [114,127,141]. When carrying out these activities, having a systematic approach is beneficial that can assist ontology developers to identify the quality characteristics and measures relevant to the intended needs. Then, to select the appropriate evaluation methods and tools for assessing characteristics. Based on our survey, a few contributions that cover a part of the ontology quality assessment process were identified. For instance, ROMEO methodology provides a set of guidelines for specifying intrinsic quality characteristics and measures after identifying the ontology extrinsic requirements [150]. The authors in [15] have discussed the specification of measures associated with the selected ontology characteristics such as accuracy, completeness, consistency and uniqueness (i.e., conciseness). Moreover, there are several evaluation approaches (i.e., application-based, data-driven, golden standards and expert-based) as presented in survey papers [4,22,112,154]. These approaches discuss the techniques of ontology evaluation that usually can be performed after ontology development. In addition to that, ontology methodologies and design patterns provide guidelines for modeling well-designed ontologies as discussed in Section 3. However, none of the methodologies discusses quality specifications in relation to intended needs and their evaluation in detail.

Due to the lack of a systematic approach that covers ontology quality specification and evaluation, it is difficult for inexperienced developers to identify a proper set of characteristics related to the considered use cases and to select appropriate methods and tools for evaluation [4,18,114,147]. Thus, in turn, the following pitfalls can occur (i) quality evaluation is limited to the frequently discussed characteristics such as functional completeness (i.e., expressiveness), consistency, and practical usefulness [32], and in turn (ii) essential quality characteristics which are required for a considered context of use in the domain may get ignored [32,114,147] and (iii) inappropriate evaluation approaches/methods and tools for a considered context can be selected. To this end, the derived conceptual quality model through this survey helps to provide an understanding of the possible characteristics associated with ontology evaluation aspects (i.e., intrinsic and extrinsic). Thereby, it supports preventing pitfalls (i) and (ii). However, it is worth having a set of formal guidelines that supports ontologists to systematically specify the extrinsic and intrinsic characteristics of an ontology in relation to the intended needs of the system that the ontology is integrated with [102]. Thereafter, to identify the appropriate evaluation methods and tools for a considered use case.

7.3. *Lack of detailed guidelines for selecting the correct modeling choices at the conceptual level*

As explained in Section 3 (i.e., Ontology development methodologies and design patterns), the effectiveness of the steps taken to construct an ontology could potentially have an impact on the quality of the ontology. Particularly, the *modeling choices* that ontology developers choose during the ontology design phase affect the ontology quality. If simple examples for *modeling choices* in ontology development are provided, a *modeling choice* might be deciding whether to represent a concept as a class or as an individual [17,64]. Another *modeling choice* could be determining the appropriate relationships between different concepts, such as “part of” or “subclass of” [17,64]. These decisions depend on the intended use of the ontology, the domain of application, and the available resources and knowledge. However, if these modeling choices are not properly selected to model the ontology, then ontology will become inconsistent either extrinsically or intrinsically [3,20,79]. This issue has further been discussed in [20]. For example, the authors in [79] have explained the problems that occurred in applying inappropriate modeling choices when developing the Data Mining OPtimization Ontology (DMOP) and how the problems were resolved using good modeling choices. The authors in [3] explain how the incorrect semantic association among classes was solved using a well-founded modeling solution introduced in OntoUML. Furthermore, the study of Terkaj et al. [134] demonstrates how several modeling options can be employed depending on the engineering requirements in the manufacturing domain. Although there are research studies like these to understand the good modeling options related to the selected use cases, their recommendations are only applicable to a few domains. Not only that, it is important to assess the chosen modeling options to make sure they produce good-quality ontologies. In this situation, it is useful to have guidelines for selecting the best modeling choices [3]. However, producing such guidelines is challenging and requires analysis of how the various modeling options have been used in successful use cases. Moreover, it is also feasible to examine the benefits and drawbacks of using those modeling options and discuss their effect on producing a quality ontology. This remains an open topic that interested researchers can further investigate.

7.4. Lack of methods and tools for assessing the extrinsic quality

The use of methods and tools makes the ontology evaluation process easy and reduces the cost of manual evaluation [27,154]. Based on the analysis of the related survey papers and literature reviews [7,27,66,105,111,154], it has been observed that several methods and tools have been introduced to support ontology quality assessment. Table 13 summarizes a set of tools/methods and the evaluation aspects that each tool/method has covered. Based on that, it can be understood that limited work has been carried out related to the extrinsic aspect. Moreover, the survey performed in [105] analyzed thirteen tools. The results of that survey [105] also have shown that a few tools such as OntologyTest [52], COLORE [63], and Open Link Virtuoso support [119] assess the domain extrinsic characteristics/attributes. None of the selected tools in that survey has focused on the application extrinsic aspect. Specifically, it has been highlighted that the functions: assessing query time performance (i.e., efficiency), validating the application requirements of the system that the ontology is integrated in, and assessing user experience with ontologies (i.e., quality in use) have not been considered. Accordingly, it is evident that introducing tools and methods for extrinsic quality evaluation are still open for future research.

8. Conclusion and future works

The findings of a systematic review of ontology quality assessments were reported in this article. The review considered thirty (30) papers published from 2010 to 2020 and six (06) milestone papers published before 2010 including four (04) other significant papers. As a result, around forty (40) papers were reviewed in total (see the Appendix). *The main contribution of this article is that we proposed a quality model for ontology quality assessment (see Table 6) with formalized definitions of the characteristics and related measures. Nineteen characteristics were discovered primarily in relation to the four aspects of the ontology evaluation space. Out of these nineteen characteristics,* timeliness *was linked to two additional characteristics:* currentness *and* volatility. *Thus, altogether twenty-one definitions were derived*. The proposed quality model (see Table 6) would provide an underpinning for ontology quality assessment and further experiments are required for a more complete model with a balanced set of characteristics, thereby it can be adapted for any domain with minimum amendments. Additionally, it is vital to empirically explore the effect (i.e., positive, negative) of changes in one characteristic to another that has not been so far discussed. Instead of that, currently, researchers have made assumptions about the correlation between the characteristics that cannot be acknowledged without rigorous experiments as they can be varied in the context where the ontology is built for. Based on the comparison of the previous works, it can be observed that none of the quality models and approaches covered all characteristics in the ontology aspects, nevertheless, a wide range of characteristics have been discussed in OQuaRE [40]. However, it does not support evaluating the semantic features of an ontology and the proposed attributes are subjective. Moreover, there is limited evidence related to the quality evaluation in the extrinsic aspect of ontologies, thus, more research on the extrinsic edge is required. In the next step, the aim is to empirically evaluate a set of ontologies that are modeled for a specified task with the identified characteristics and afterward, to propose a systematic approach for ontology quality specification and evaluation concerning a use-case in a selected domain.

Acknowledgement

The author acknowledges the support received from the LK Domain Registry in publishing this paper, https://www.domains.lk/outreach/research-grants/.

Table 13

Preliminary analysis of methods and tools

Tool/Method	Topic	Characteristics/Attributes	Aspects
RDF Validation Service [144]	Tool: syntax validator	Language compliance	Structural Intrinsic
OWL Validator [73]	Tool: syntax validator	Language compliance	Structural Intrinsic
OntoAnalyser [58,126]	Plug-in (i.e., OntoEdit): for language conformity and consistency analysis	Compliance and Internal Consistency	Structural Intrinsic
OntoKick [126]	Plug-in (i.e., OntoEdit): requirements specification and evaluation	Accuracy and functional completeness	Domain Extrinsic
OntoClean [65]	Method: for validating the ontological adequacy of taxonomic relationships	External Consistency	Domain Intrinsic
OntoQA [132]	Tool: for metric-based ontology quality analysis	Complexity: relationship richness, attribute richness, inheritance richness, class richness, Average population, fullness. Modularity: cohesion, importance, connectivity, instance relationship richness, etc.	Structural Intrinsic
Unit Test [143]	Method: for analyzing unwanted changes and side effects during the maintenance	External Consistency and Accuracy	Domain Intrinsic and Domain Extrinsic
S-OntoEval tool [37]	Tool: ontology quality analysis	Complexity, Modularity, Internal Consistency, External Consistency, Comprehensibility	Structural Intrinsic and Domain Intrinsic
OntologyTest [52]	Tool: for checking ontology functional requirements	Accuracy and Functional completeness	Domain Extrinsic
OntoCheck [118]	Plug-in (i.e., Protégé): for verifying annotations and naming conventions	Comprehensibility	Structural Intrinsic attributes that are complementary to the domain intrinsic aspect have been automated.
XD analyzer [30]	Plug-in (i.e., NeOn Toolkit): for verifying annotations and naming conventions	Coverage: isolated entities, missing types, missing domain or range in properties, missing inverse, Comprehensibility: instance missing labels and comments, unused imported ontologies	Structural Intrinsic attributes that are complementary to the domain intrinsic aspect have been automated.
Copeland et al. method [26]	Method: for ontology regression test	Internal consistency and External Consistency	Structural Intrinsic and Domain Intrinsic
RepOSE[a] [89]	Tool: for detecting and repairing defects in ontologies and alignments	Compliance and External Consistency	Structural intrinsic and Domain Intrinsic

Table 13

(Continued)

Tool/Method	Topic	Characteristics/Attributes	Aspects
OOPs[b] [111]	Tool: for common pitfalls detection (i.e., pitfalls have been numbered as P01, P02, … P041 in [98])	Compliance: e.g., P34, P35 and P38. Consistency: e.g., P05, P06, P07, P19 and P24. Coverage: e.g., P04, P10, P11, P12 and P13. Conciseness: e.g., P02, P03, P21 and P32. Comprehensibility: e.g., P08, P20 and P22. Availability: e.g., P36 and P37.	Structural Intrinsic, Domain Intrinsic/Extrinsic and Application Extrinsic (i.e., only automated structural intrinsic metrics)
OntoMetric[c] [90,91]	Tool: for metric-based ontology quality analysis	Complexity: basic metric, knowledge base metric Modularity: graph metrics, class metrics Comprehensibility: annotation metrics	Structural Intrinsic and Domain Intrinsic (i.e., only automated structural intrinsic metrics)
OntoDebug [117]	Plug-in (i.e., Protégé): for ontology inconsistency debugging	Internal Consistency	Structural Intrinsic
DoORS[d] [98]	Tool: metric-based ontology quality analysis	Compliance, External Consistency, Conciseness, Comprehensibility, Accuracy, Relevancy, and Credibility	Structural Intrinsic, Domain Intrinsic and Domain Extrinsic
OntoKeeper [5]	Tool: for metric-based ontology quality analysis	Compliance: lawfulness, richness, Conciseness, Comprehensibility, Accuracy, Relevancy, and Credibility	Structural Intrinsic, Domain Intrinsic, and Domain Extrinsic
Delta [86]	Tool: for ontology quality analysis (i.e., adopted OntoQA metrics and used OOPs as an external plug-in)	Complexity, Modularity	Structural Intrinsic

[a] RepOSE: https://www.ida.liu.se/~patla00/research/RepOSE/

[b] OOPS!: https://oops.linkeddata.es/

[c] OntoMetric: https://ontometrics.informatik.uni-rostock.de/ontologymetrics/

[d] DoORS: https://owlparser.herokuapp.com/

Appendix

Table A.1

The core papers considered for the survey

Citation	Year	Title of the paper
The papers selected upon the inclusion and exclusion criteria defined in the systematic review (Core Survey Papers)		
Evermann et al. [41]	2010	Evaluating ontologies: Towards a cognitive measure of quality
Ma et al. [94]	2010	Semantic oriented ontology cohesion metrics for ontology-based systems
Djedidi et al. [38]	2010	ONTO-EVO^{A}L an Ontology Evolution Approach Guided by Pattern Modeling and Quality Evaluation
Tartir et al. [133]	2010	Ontological evaluation and validation
Oh et al. [104]	2011	Cohesion and coupling metrics for ontology modules
Duque-Ramos et al. [40]	2011	OQuaRE: A SQuaRE-based Approach for Evaluating the Quality of Ontologies
Murdock et al. [100]	2011	Evaluating Dynamic Ontologies
Bouiadjra et al. [10]	2011	FOEval: Full Ontology Evaluation
Ouyang et al. [107]	2011	A Method of Ontology Evaluation Based on Coverage, Cohesion and Coupling
Gavrilova et al. [54]	2012	New Ergonomic Metrics for Educational Ontology Design and Evaluation
Schober et al. [118]	2012	OntoCheck: verifying ontology naming conventions and metadata completeness in Protégé 4
Haghighi et al. [34]	2013	Development and evaluation of ontology for intelligent decision support in medical emergency management for mass gatherings
Neuhaus et al. [102]	2013	Towards ontology evaluation across the life cycle
Sánchez et al. [116]	2014	Semantic variance: An intuitive measure for ontology accuracy evaluation
Poveda-Villalón et al. [111]	2014	OOPS! (OntOlogy Pitfall Scanner!): An On-line Tool for Ontology Evaluation
Alexopoulos et al. [2]	2014	Towards Vagueness-Oriented Quality Assessment of Ontologies
Rico et al. [114]	2014	OntoQualitas: A framework for ontology quality assessment in information interchanges between heterogeneous systems
Batet et al. [12]	2014	A Semantic Approach for Ontology Evaluation
Radulovic et al. [113]	2015	SemQuaRE – An extension of the SQuaRE quality model for the evaluation of semantic technologies
Lantow [91]	2016	OntoMetrics: Putting Metrics into Use for Ontology Evaluation
McDaniel et al. [97]	2016	The Role of Community Acceptance in Assessing Ontology Quality
Hnatkowska et al. [72]	2017	Verification of SUMO ontology
Zhu et al. [153]	2017	Quality Model and Metrics of Ontology for Semantic Descriptions of Web Services
Tan et al. [130]	2017	Evaluation of an Application Ontology
Ashraf et al. [8]	2018	Evaluation of U Ontology
McDaniel et al. [98]	2018	Assessing the Quality of Domain Ontologies: Metrics and an Automated Ranking System
Kumar et al. [88]	2018	Quality Evaluation of Ontologies
Demaidi et al. [36]	2019	TONE: A Method for Terminological Ontology Evaluation
Amith et al. [5]	2019	Architecture and usability of OntoKeeper, an ontology evaluation tool
Franco et al. [45]	2020	Evaluation of ontology structural metrics based on public repository data
Other supported papers including milestone papers published before 2010		
Gruninger et al. [64]	1995	Methodology for Design and Evaluation of Ontologies
Gomez-Perez [58]	2001	Evaluation of ontologies
Burton-Jones et al. [24]	2005	A semiotic metrics suite for assessing the quality of ontologies
Haase et al. [68]	2005	Consistent Evolution of OWL Ontologies
Gangemi et al. [49]	2006	Modelling Ontology Evaluation and Validation
Stvilia [124]	2007	A model for ontology quality evaluation
Tsarkov [137]	2012	Improved algorithms for module extraction and atomic decomposition
Vescovo et al. [33]	2013	Empirical study of logic-based modules: Cheap is cheerful
Khan et al. [81]	2015	An empirically-based framework for ontology modularization
Bayoudhi et al. [14]	2018	How to Repair Inconsistency in OWL 2 DL Ontology Versions?

References

[1] A.S. Abdelghany, N.R. Darwish and H.A. Hefni, An agile methodology for ontology development, *International Journal of Intelligent Engineering and Systems* **12**(2) (2019), 170–181. doi:10.22266/ijies2019.0430.17.

[2] P. Alexopoulos and P. Mylonas, Towards vagueness-oriented quality assessment of ontologies, in: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, 2014, pp. 448–453. doi:10.1007/978-3-319-07064-3_38.

[3] G. Amaral, F. Baião and G. Guizzardi, Foundational ontologies, ontology-driven conceptual modeling, and their multiple benefits to data mining, *WIREs Data Mining Knowl Discov.* **11** (2021). doi:10.1002/widm.1408.

[4] M. Amith, Z. He, J. Bian, J.A. Lossio-Ventura and C. Tao, Assessing the practice of biomedical ontology evaluation: Gaps and opportunities, *Journal of Biomedical Informatics* **80** (2018), 1–13. doi:10.1016/j.jbi.2018.02.010.

[5] M. Amith, F. Manion, C. Liang, M. Harris, D. Wang, Y. He and C. Tao, Architecture and usability of OntoKeeper, an ontology evaluation tool, *BMC Med Inform Decis Mak.* **19** (2019), 152. doi:10.1186/s12911-019-0859-z.

[6] M.E. Aranguren, E. Antezana, M. Kuiper and R. Stevens, Ontology design patterns for bio-ontologies: A case study on the cell cycle ontology, *BMC Bioinformatics* **9** (2008), S1. doi:10.1186/1471-2105-9-S5-S1.

[7] T. Aruna, K. Saranya and C. Bhandari, A survey on ontology evaluation tools, in: *2011 International Conference on Process Automation, Control and Computing*, 2011, pp. 1–5. doi:10.1109/PACC.2011.5978931.

[8] J. Ashraf, O.K. Hussain, F.K. Hussain and E.J. Chang, Evaluation of u ontology, in: *Measuring and Analysing the Use of Ontologies*, Springer, 2018, pp. 243–268. doi:10.1007/978-3-319-75681-3_9.

[9] Association for Ontology Design & Patterns (ODPA), OPTypes – Odp, Ontology Design Patterns. Org (ODP), 2021, http://ontologydesignpatterns.org/wiki/OPTypes (accessed December 29, 2021).

[10] A. Bachir Bouiadjra and S. Benslimane, FOEval: Full ontology evaluation, in: *2011 7th International Conference on Natural Language Processing and Knowledge Engineering*, 2011, pp. 464–468. doi:10.1109/NLPKE.2011.6138244.

[11] V.R. Basili, G. Caldiera and H.D. Rombach, The goal question metric approach, *Encyclopedia of Software Engineering* **1** (1994), 528–532.

[12] M. Batet and D. Sánchez, A semantic approach for ontology evaluation, in: *IEEE 26th International Conference on Tools with Artificial Intelligence*, 2014, pp. 138–145. doi:10.1109/ICTAI.2014.30.

[13] C. Batini and M. Scannapieco, *Data Quality: Concepts, Methodologies and Techniques*, Springer-Verlag, New York, 2006. doi:10.1007/3-540-33173-5.

[14] L. Bayoudhi, N. Sassi and W. Jaziri, How to repair inconsistency in OWL 2 DL ontology versions?, *Data & Knowledge Engineering* **116** (2018), 138–158. doi:10.1016/j.datak.2018.05.010.

[15] B. Behkamal, M. Kahani, E. Bagheri and Z. Jeremic, A metrics-driven approach for quality assessment of linked open data, *Journal of Theoretical and Applied Electronic Commerce Research* **9** (2014), 16. doi:10.4067/S0718-18762014000200006.

[16] A.B. Benevides and G. Guizzardi, A model-based tool for conceptual modeling and domain ontology engineering in OntoUML, in: *Enterprise Information Systems*, J. Filipe and J. Cordeiro, eds, Springer, Berlin, Heidelberg, 2009, pp. 528–538. doi:10.1007/978-3-642-01347-8_44.

[17] N. Bevan, Usability is quality of use, in: *Advances in Human Factors/Ergonomics*, Elsevier, 1995, pp. 349–354. doi:10.1016/S0921-2647(06)80241-8.

[18] E. Blomqvist, A. Gangemi and V. Presutti, Experiments on pattern-based ontology design, in: *Proceedings of the Fifth International Conference on Knowledge Capture*, Association for Computing Machinery, New York, NY, USA, 2009, pp. 41–48. doi:10.1145/1597735.1597743.

[19] B. Boehm, J. Brown and M. Lipow, Quantitative evaluation of software quality, in: *Proceedings of the 2nd International Conference on Software Engineering*, San Francisco, USA, 1976, pp. 592–605, https://dl.acm.org/doi/10.5555/800253.807736.

[20] S. Borgo, A. Galton and O. Kutz, Foundational ontologies in action: Understanding foundational ontology through examples, *AO* **17** (2022), 1–16. doi:10.3233/AO-220265.

[21] W. Borst, Construction of Engineering Ontologies, PhD thesis, Institute for Telematica and Information Technology, University of Twente, Enschede, The Netherlands, 1997.

[22] J. Brank, M. Grobelnik and D. Mladeni, A survey of ontology evaluation techniques, in: *Proc. Conf. Data Min. Data Warehouses*, 2005, pp. 166–170.

[23] P. Brereton, B.A. Kitchenham, D. Budgen, M. Turner and M. Khalil, Lessons from applying the systematic literature review process within the software engineering domain, *Journal of Systems and Software* **80** (2007), 571–583. doi:10.1016/j.jss.2006.07.009.

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

[25] J. Cavano and J. McCall, A framework for the measurement of software quality, *ACM SIGMETRICS Performance Evaluation Review* **7** (1978), 133–139. doi:10.1145/953579.811113.

[26] M. Copeland, R.S. Gonçalves, B. Parsia, U. Sattler and R. Stevens, Finding fault: Detecting issues in a versioned ontology, in: *The Semantic Web: ESWC 2013 Satellite Events*, P. Cimiano, M. Fernández, V. Lopez, S. Schlobach and J. Völker, eds, Springer, Berlin, Heidelberg, 2013, pp. 113–124. doi:10.1007/978-3-642-41242-4_10.

[27] O. Corcho, M. Fernández-López and A. Gómez-Pérez, Methodologies, tools and languages for building ontologies. Where is their meeting point?, *Data & Knowledge Engineering* **46** (2003), 41–64. doi:10.1016/S0169-023X(02)00195-7.

[28] Credibility, https://dictionary.cambridge.org/dictionary/english/credibility (accessed June 26, 2021).

[29] M. Cristani and R. Cuel, A survey on ontology creation methodologies, *International Journal on Semantic Web and Information Systems (IJSWIS)* **2** (2005), 49–69. doi:10.4018/jswis.2005040103.

[30] E. Daga, XDTools – NeOn Wiki, 2012, http://neon-toolkit.org/wiki/XDTools.html (accessed December 29, 2021).

[31] J. Defeo, *Juran's Quality Handbook: The Complete Guide to Performance Excellence*, 7th edn, McGraw Hill, New York, 2016.

[32] A. Degbelo, A snapshot of ontology evaluation criteria and strategies, in: *Proceedings of the 13th International Conference on Semantic Systems (Semantics2017)*, Association for Computing Machinery, New York, NY, USA, 2017, pp. 1–8. doi:10.1145/3132218.3132219.

[33] C. Del Vescovo, P. Klinov, B. Parsia, U. Sattler, T. Schneider and D. Tsarkov, Empirical study of logic-based modules: Cheap is cheerful, in: *Proceedings of the 26th International Workshop on Description Logics (DL'13), CEUR Workshop Proceedings*, Ulm, Germany, 2013, pp. 144–155. doi:10.1007/978-3-642-41335-3_6.

[34] P. Delir Haghighi, F. Burstein, A. Zaslavsky and P. Arbon, Development and evaluation of ontology for intelligent decision support in medical emergency management for mass gatherings, *Decision Support Systems* **54** (2013), 1192–1204. doi:10.1016/j.dss.2012.11.013.

[35] W.H. DeLone and E.R. McLean, The DeLone and McLean model of information systems success: A ten-year update, *J. Manag. Inf. Syst.* **19** (2003), 9–30. doi:10.1080/07421222.2003.11045748.

[36] M.N. Demaidi and M.M. Gaber, TONE: A method for terminological ontology evaluation, in: *Proceedings of the ArabWIC 6th Annual International Conference Research Track*, Association for Computing Machinery, New York, NY, USA, 2019, pp. 1–10. doi:10.1145/3333165.3333179.

[37] R.Q. Dividino, M. Romanelli and D. Sonntag, Semiotic-based ontology evaluation tool (S-OntoEval), in: *LREC*, 2008, http://www.lrec-conf.org/proceedings/lrec2008/pdf/671_paper.pdf.

[38] R. Djedidi and M.A. Aufaure, ONTO-EVO A L an ontology evolution approach guided by pattern modeling and quality evaluation, in: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010, pp. 286–305. doi:10.1007/978-3-642-11829-6_19.

[39] R. Dromey, *Software Product Quality: Theory, Model, and Practice*, Software Quality Institute, Brisbane, Australia, 1998.

[40] A. Duque-Ramos, J.T. Fernández-Breis, R. Stevens and N. Aussenac-Gilles, OQuaRE: A SQuaRE-based approach for evaluating the quality of ontologies, *Journal of Research and Practice in Information Technology* **43** (2011), 159–176, https://search.informit.org/doi/10.3316/ielapa.265844843145749.

[41] J. Evermann and J. Fang, Evaluating ontologies: Towards a cognitive measure of quality, *Information Systems* **35** (2010), 391–403. doi:10.1016/j.is.2008.09.001.

[42] M. Fernández, C. Overbeeke, M. Sabou and E. Motta, What makes a good ontology? A case-study in fine-grained knowledge reuse, in: *The Semantic Web*, 2009. doi:10.1007/978-3-642-10871-6_5.

[43] G. Flouris, D. Manakanatas, H. Kondylakis, D. Plexousakis and G. Antoniou, Ontology change: Classification and survey, *Knowledge Eng. Review* **23** (2008), 117–152. doi:10.1017/S0269888908001367.

[44] M.S. Fox and M. Grüninger, Ontologies for enterprise modelling, in: *Enterprise Engineering and Integration. Research Reports Esprit*, K. Kosanke and J.G. Nell, eds, Springer, Berlin, Heidelberg, 1997, pp. 190–200. doi:10.1007/978-3-642-60889-6_22.

[45] M. Franco, J.M. Vivo, M. Quesada-Mart'ınez, A. Duque-Ramos and J.T. Fernández-Breis, Evaluation of ontology structural metrics based on public repository data, *Briefings in Bioinformatics* **21** (2020), 473–485. doi:10.1093/bib/bbz009.

[46] C. Fürber and M. Hepp, SWIQA – a semantic web information quality assessment framework, in: *CIS 2011 Proceedings*, 2011, p. 76, https://aisel.aisnet.org/ecis2011/76.

[47] A. Gangemi, Ontology design patterns for Semantic Web content, in: The *Semantic Web – ISWC 2005*, Y. Gil, E. Motta, V.R. Benjamins and M.A. Musen, eds, Springer, Berlin, Heidelberg, 2005, pp. 262–276. doi:10.1007/11574620_21.

[48] A. Gangemi, C. Catenacci, M. Ciaramita and J. Lehmann, A theoretical framework for ontology evaluation and validation, in: *Semantic Web Applications and Perspectives (SWAP) – 2nd Italian Semantic Web Workshop*, Trento, Italy, 2005, p. 16.

[49] A. Gangemi, C. Catenacci, M. Ciaramita and J. Lehmann, Modelling ontology evaluation and validation, in: *The Semantic Web: Research and Applications*, Y. Sure and J. Domingue, eds, Springer, Berlin, Heidelberg, 2006, pp. 140–154. doi:10.1007/11762256_13.

[50] A. Gangemi, C. Catenacci, M. Ciaramita, J. Lehmann, R. Gil, F. Bolici and O. Strignano, Ontology evaluation and validation: An integrated formal model for the quality diagnostic task, Technical report, Lab. for Applied Ontology, 2005.

[51] J. García, J. García-Peñalvo and R. Therón, A survey on ontology metrics, in: *Knowledge Management, Information Systems, E-Learning, and Sustainability Research. WSKS 2010*, M.D. Lytras, P. Ordonez De Pablos, A. Ziderman, A. Roulstone, H. Maurer and J.B. Imber, eds, Communications in Computer and Information Science, Springer, Berlin, Heidelberg, 2010, pp. 22–27. doi:10.1007/978-3-642-16318-0_4.

[52] S. García-Ramos, A. Otero and M. Fernández-López, OntologyTest: A tool to evaluate ontologies through tests defined by the user, in: *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*, S. Omatu, M.P. Rocha, J. Bravo, F. Fernández, E. Corchado, A. Bustillo and J.M. Corchado, eds, Springer, Berlin, Heidelberg, 2009, pp. 91–98. doi:10.1007/978-3-642-02481-8_13.

[53] D. Garvin, What does 'product quality' really mean? in: *Sloan Management Review*, Fall, 1984, pp. 25–45.

[54] T. Gavrilova, V. Gorovoy and E. Bolotnikova, New ergonomic metrics for educational ontology design and evaluation, *Frontiers in Artificial Intelligence and Applications* **246** (2012), 361–378. doi:10.3233/978-1-61499-125-0-361.

[55] A. Gillies, *Software Quality: Theory and Management*, Chapman & Hall, London, UK, 1992, p. 250. doi:10.1002/stvr.4370020305.

[56] A. Gomez-Perez, Some ideas and examples to evaluate ontologies, in: *Proceedings the 11th Conference on Artificial Intelligence for Applications*, IEEE Comput. Soc. Press, Los Angeles, CA, USA, 1995, pp. 299–305. doi:10.1109/CAIA.1995.378808.

[57] A. Gómez-Pérez, Towards a framework to verify knowledge sharing technology, *Expert Systems with Applications* **11** (1996), 519–529. doi:10.1016/S0957-4174(96)00067-X.

[58] A. Gomez-Perez, Evaluation of ontologies, *Int. J. Intell. Syst.* **16** (2001), 391–409. doi:10.1002/1098-111X(200103)16:3<391::AID-INT1014>3.0.CO;2-2.

[59] A. Gomez-Perez, M. Fernandez-Lopez and A. de Vicente, Towards a method to conceptualize domain ontologies, in: *ECAI96 Workshop on Ontological Engineering*, Budapest, 1996, pp. 41–51, https://oa.upm.es/7228/.

[60] B.C. Grau, B. Motik, G. Stoilos and I. Horrocks, Completeness guarantees for incomplete ontology reasoners: Theory and practice, *J. Artif. Int. Res.* **43** (2012), 419–476. doi:10.1613/jair.3470.

[61] T.R. Gruber, A translation approach to portable ontology specifications, *Knowledge Acquisition* **5** (1993), 199–220. doi:10.1006/knac.1993.1008.

[62] T.R. Gruber, Toward principles for the design of ontologies used for knowledge sharing, *International Journal of Human Computer Studies* **43** (1995), 907–928. doi:10.1006/ijhc.1995.1081.

[63] M. Grüninger, Ontology evaluation workflow in COLORE, 2013, http://ontolog.cim3.net/file/work/OntologySummit2013/2013-02-14_OntologySummit2013_OntologyEvaluation-SoftwareEnvironment/wip/Gruninger_colore-summit_20130214a.pdf.

[64] M. Gruninger and M.S. Fox, Methodology for design and evaluation of ontologies, in: *Workshop on Basic Ontological Issues in Knowlege Sharing*, 1995, http://www.eil.utoronto.ca/wp-content/uploads/enterprise-modelling/papers/gruninger-ijcai95.pdf (accessed September 21, 2021).

[65] N. Guarino and C.A. Welty, An overview of OntoClean, in: *Handbook on Ontologies*, S. Staab and R. Studer, eds, Springer, Berlin, Heidelberg, 2004, pp. 151–171. doi:10.1007/978-3-540-24750-0_8.

[66] A. Gyrard, G. Atemezing and M. Serrano, PerfectO: An online toolkit for improving quality, accessibility, and classification of domain-based ontologies, in: *Semantic IoT: Theory and Applications: Interoperability, Provenance and Beyond*, 2021, pp. 161–192. doi:10.1007/978-3-030-64619-6_7.

[67] P. Haase, F.V. Harmelen, Z. Huang, H. Stuckenschmidt and Y. Sure, A framework for handling inconsistency in changing ontologies, in: *The Semantic Web – ISWC 2005*, Y. Gil, E. Motta, V.R. Benjamins and M.A. Musen, eds, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2005, pp. 353–367. doi:10.1007/11574620_27.

[68] P. Haase and L. Stojanovic, Consistent evolution of OWL ontologies, in: *The Semantic Web: Research and Applications*, A. Gómez-Pérez and J. Euzenat, eds, Springer, Berlin, Heidelberg, 2005, pp. 182–197. doi:10.1007/11431053_13.

[69] K. Hammar, *Content Ontology Design Patterns: Qualities, Methods, and Tools*, Linköping University Electronic Press, 2017. doi:10.3384/diss.diva-139584.

[70] J. Hartmann, P. Spyns, A. Giboin, D. Maynard, R. Cuel and M.C. Suárez-Figueroa, D1.2.3 Methods for ontology evaluation, 2005, EU-IST Network of Excellence (NoE) IST-2004-507482 KWEB Deliverable D1.2.3 (WP 1.2).

[71] P. Hitzler, Anti-patterns in ontology-driven conceptual modeling: The case of role modeling in OntoUML. Ontology engineering with ontology design patterns, *Foundations and Applications* **25** (2016), 161.

[72] B. Hnatkowska, Verification of SUMO ontology, *Journal-of-intelligent-and-fuzzy-systems/ifs169118* **32** (2017), 1183–1192. doi:10.3233/JIFS-169118.

[73] M. Horridge, OWL 2 Validator, 2009, http://mowl-power.cs.man.ac.uk:8080/validator/ (accessed December 29, 2021).

[74] IEEE 730-2014 – IEEE Standard for Software Quality Assurance Processes, https://standards.ieee.org/standard/730-2014.html (accessed June 26, 2021).

[75] ISO 25012, https://iso25000.com/index.php/en/iso-25000-standards/iso-25012 (accessed June 26, 2021).

[76] ISO 9241-11:2018(en), Ergonomics of human-system interaction – Part 11: Usability: Definitions and concepts, https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en (accessed June 26, 2021).

[77] ISO/IEC 25010:2011(en), Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models, https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en (accessed June 26, 2021).

[78] ISO/IEC TR 9126-4:2004(en), Software engineering – Product quality – Part 4: Quality in use metrics, https://www.iso.org/obp/ui/fr/#iso:std:iso-iec:tr:9126:-4:ed-1:v1:en (accessed June 26, 2021).

[79] C.M. Keet and A. Ławrynowicz, Modeling issues and choices in the Data Mining OPtimization Ontology, 2013, http://researchspace.csir.co.za/dspace/bitstream/handle/10204/7390/Keet2_2013.pdf?sequence=1.

[80] D.D. Kehagias, I. Papadimitriou, J. Hois, D. Tzovaras and J. Bateman, A Methodological Approach for Ontology Evaluation and Refinement, 2008, p. 12.

[81] Z.C. Khan and C.M. Keet, An empirically-based framework for ontology modularisation, *Applied Ontology* **10** (2015), 171–195. doi:10.3233/AO-150151.

[82] M.R. Khondoker and P. Mueller, Comparing ontology development tools based on an online survey, in: *World Congress on Engineering (WCE 10)*, London, UK, 2010.

[83] B. Kitchenham, Procedures for undertaking systematic reviews, Joint Technical Report, Computer Science Department, Keele University (TR/SE0401) and National ICT Australia Ltd. (0400011T.1), 2004.

[84] B. Kitchenham, S.L. Pfleeger and N. Fenton, Towards a framework for software measurement validation, *IIEEE Trans. Software Eng.* **21** (1995), 929–944. doi:10.1109/32.489070.

[85] M. Klein, D. Fensel, A. Kiryakov and D. Ognyanov, Ontology versioning and change detection on the web, in: *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web. EKAW 2002*, A. Gómez-Pérez and V.R. Benjamins, eds, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2002, pp. 197–212. doi:10.1007/3-540-45810-7_20.

[86] H. Kondylakis, A. Nikolaos, P. Dimitra, K. Anastasios, K. Emmanouel, K. Kyriakos, S. Iraklis, K. Stylianos and N. Papadakis, A modular ontology evaluation system, *Information* **12** (2021), 301. doi:10.3390/info12080301.

[87] K.I. Kotis, G.A. Vouros and D. Spiliotopoulos, Ontology engineering methodologies for the evolution of living and reused ontologies: Status, trends, findings and recommendations, *The Knowledge Engineering Review* **35** (2020), e4. doi:10.1017/S0269888920000065.

[88] S. Kumar and N. Baliyan, Quality evaluation of ontologies, in: *Semantic Web-Based Systems*, Springer, Singapore, Singapore, 2018, pp. 19–50. doi:10.1007/978-981-10-7700-5_2.

[89] P. Lambrix and Q. Liu, Debugging the missing is-a structure within taxonomies networked by partial reference alignments, *Data & Knowledge Engineering* **86** (2013), 179–205. doi:10.1016/j.datak.2013.03.003.

[90] B. Lantow, OntoMetrics: Application of on-line ontology metric calculation, in: *BIR Workshops*, 2016, pp. 1–12.

[91] B. Lantow, OntoMetrics: Putting metrics into use for ontology evaluation, in: *KEOD*, 2016, pp. 186–191. doi:10.5220/0006084601860191.

[92] D.B. Lenat and R.V. Guha, *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*, Addison-Wesley, Boston, 1990. doi:10.1016/0004-3702(93)90092-P.

[93] R. Lourdusamy and A. John, A review on metrics for ontology evaluation, in: *2nd International Conference on Inventive Systems and Control (ICISC)*, 2018, pp. 1415–1421. doi:10.1109/ICISC.2018.8399041.

[94] Y. Ma, B. Jin and Y. Feng, Semantic oriented ontology cohesion metrics for ontology-based systems, *Journal of Systems and Software* **83** (2010), 143–152. doi:10.1016/j.jss.2009.07.047.

[95] S. Mc Gurk, C. Abela and J. Debattista, Towards Ontology Quality Assessment, in: *MEPDaW/LDQ@ ESWC*, 2017, pp. 94–106.

[96] M. McDaniel and V.C. Storey, Evaluating domain ontologies: Clarification, classification, and challenges, *ACM Comput. Surv.* **52** (2019). doi:10.1145/3329124.

[97] M. McDaniel, V.C. Storey and V. Sugumaran, The role of community acceptance in assessing ontology quality, in: *Natural Language Processing and Information Systems*, E. Métais, F. Meziane, M. Saraee, V. Sugumaran and S. Vadera, eds, Lecture Notes in Computer Science, Springer, Cham, 2016, pp. 24–36. doi:10.1007/978-3-319-41754-7_3.

[98] M. McDaniel, V.C. Storey and V. Sugumaran, Assessing the quality of domain ontologies: Metrics and an automated ranking system, *Data & Knowledge Engineering* **115** (2018), 32–47. doi:10.1016/j.datak.2018.02.001.

[99] B. Motik, P.F. Patel and B. Parsia, *OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax*, 2nd edn, 2012, https://www.w3.org/TR/owl2-syntax/ (accessed December 29, 2021).

[100] J. Murdock, C. Buckner and C. Allen, Evaluating dynamic ontologies, in: *Communications in Computer and Information Science*, Springer Verlag, 2013, pp. 258–275. doi:10.1007/978-3-642-29764-9_18.

[101] F. Neuhaus, S. Ray and R.D. Sriram, National Institute of Standards and Technology, U.S. Department of Commerce, 2014, https://nvlpubs.nist.gov/nistpubs/ir/2014/NIST.IR.8008.pdf (accessed July 23, 2021).

[102] F. Neuhaus, A. Vizedom, K. Baclawski, M. Bennett, M. Dean, M. Denny, M. Grüninger, A. Hashemi, T. Longstreth, L. Obrst, S. Ray, R. Sriram, T. Schneider, M. Vegetti, M. West and P. Yim, Towards ontology evaluation across the life cycle: The ontology summit 2013, *Applied Ontology* **8** (2013), 179–194. doi:10.3233/AO-130125.

[103] N.F. Noy and D.L. McGuinness, What is an ontology and why we need it, Ontology Development 101: A Guide to Creating Your First Ontology, https://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html (accessed June 26, 2021).

[104] S. Oh, H.Y. Yeom and J. Ahn, Cohesion and coupling metrics for ontology modules, *Information Technology and Management* **12** (2011), 81. doi:10.1007/s10799-011-0094-5.

[105] Ontology Summit, 2013, http://ontolog.cim3.net/OntologySummit/2013/ (accessed June 26, 2021).

[106] OntoUML – OntoUML specification documentation, 2018, https://ontouml.readthedocs.io/en/latest/intro/ontouml.html (accessed December 29, 2021).

[107] L. Ouyang, B. Zou, M. Qu and C. Zhang, A method of ontology evaluation based on coverage, cohesion and coupling, in: *2011 Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 2011, pp. 2451–2455. doi:10.1109/FSKD.2011.6020046.

[108] J. Pak and L. Zhou, A framework for ontology evaluation, in: *Workshop on E-Business*, 2009, pp. 10–18. doi:10.1007/978-3-642-17449-0_2.

[109] S. Peroni, A simplified agile methodology for ontology development, in: *OWL: Experiences and Directions–Reasoner Evaluation*, Springer, Cham, 2016, pp. 55–69. doi:10.1007/978-3-319-54627-8_5.

[110] H.S. Pinto, S. Staab and C. Tempich, DILIGENT: Towards a fine-grained methodology for DIstributed, *Loosely-controlled and evolvInG Engineering of oNTologies* **16** (2004), 393.

[111] M. Poveda-Villalón, A. Gómez-Pérez and M.C. Suárez-Figueroa, OOPs!(ontology pitfall scanner!): An on-line tool for ontology evaluation, *International Journal on Semantic Web and Information Systems (IJSWIS)* **10** (2014), 7–34, https://dpoi.org/10.4018/ijswis.2014040102. doi:10.4018/ijswis.2014040102.

[112] J. Raad and C. Cruz, A survey on ontology evaluation methods, in: *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, SCITEPRESS – Science and Technology Publications, Lisbon, Portugal, 2015, pp. 179–186. doi:10.5220/0005591001790186.

[113] F. Radulovic, R. García-Castro and A. Gómez-Pérez, SemQuaRE: An extension of the SQuaRE quality model for the evaluation of semantic technologies, *Computer Standards & Interfaces* **38** (2015), 101–112. doi:10.1016/j.csi.2014.09.001.

[114] M. Rico, M.L. Caliusco, O. Chiotti and M.R. Galli, OntoQualitas: A framework for ontology quality assessment in information interchanges between heterogeneous systems, *Comput. Ind.* **65** (2014), 1291–1300. doi:10.1016/j.compind.2014.07.010.

[115] C. Roussey, O. Corcho and L.M. Vilches-Blázquez, A catalogue of OWL ontology antipatterns, in: *Proceedings of the Fifth International Conference on Knowledge Capture*, Association for Computing Machinery, New York, NY, USA, 2009, pp. 205–206. doi:10.1145/1597735.1597784.

[116] D. Sánchez, Semantic variance: An intuitive measure for ontology accuracy evaluation, *Engineering Applications of Artificial Intelligence* **39** (2015), 89–99. doi:10.1016/j.engappai.2014.11.012.

[117] K. Schekotihin, P. Rodler and W. Schmid, Ontodebug: Interactive ontology debugging plug-in for protégé, in: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, 2018, pp. 340–359.

[118] D. Schober, I. Tudose, V. Svatek and M. Boeker, OntoCheck: Verifying ontology naming conventions and metadata completeness in Protégé 4, *Journal of Biomedical Semantics* **3** (2012). doi:10.1186/2041-1480-3-S2-S4.

[119] OpenLink Software, Home, Virtuoso Universal Server. (n.d.), https://www.openlinksw.com/ (accessed December 29, 2021).

[120] S. Staab, H.P. Schnurr, R. Studer and Y. Sure, Knowledge processes and ontologies, *IEEE Intelligent Systems* **16**(1) (2001), 26–34. doi:10.1109/5254.912382.

[121] S. Staab and R. Studer (eds), *Handbook on Ontologies*, 2nd edn, Springer, Berlin Heidelberg, 2009. doi:10.1007/978-3-540-24750-0.

[122] R. Stamper, K. Liu, M. Hafkamp and Y. Ades, Understanding the role of signs and norms in organizations–a semiotic approach to information systems design, *Behaviour & Information Technology* **19**(1) (2000), 15–27. doi:10.1080/014492900118768.

[123] R. Studer, V.R. Benjamins and D. Fensel, Knowledge engineering: Principles and methods, *Data & Knowledge Engineering* **25** (1998), 161–197. doi:10.1016/S0169-023X(97)00056-6.

[124] B. Stvilia, A model for ontology quality evaluation, *First Monday* **12** (2007). doi:10.5210/fm.v12i12.2043.

[125] M.C. Suárez-Figueroa, A. Gómez-Pérez and B. Villazón-Terrazas, How to write and use the ontology requirements specification document, in: *On the Move to Meaningful Internet Systems: OTM 2009*, R. Meersman, T. Dillon and P. Herrero, eds, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2009, pp. 966–982. doi:10.1007/978-3-642-05151-7_16.

[126] Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer and D. Wenke, OntoEdit: Collaborative ontology development for the semantic web, in: *The Semantic Web – ISWC 2002*, I. Horrocks and J. Hendler, eds, Springer, Berlin, Heidelberg, 2002, pp. 221–235, https://dl.acm.org/doi/10.5555/646996.711413. doi:10.1007/3-540-48005-6_18.

[127] D.W. Surry, Bringing the user into the project development process, in: *Handbook of Research on Technology Project Management, Planning, and Operations*, T.T. Kidd, ed., IGI Global, Hershey, PA, 2009, pp. 105–113. doi:10.4018/978-1-60566-400-2.ch008.

[128] B. Swartout, P. Ramesh, K. Knight and T. Russ, *Toward Distributed Use of Large-Scale Ontologies, AAAI, Symposium on Ontological Engineering*, Stanford, 1997.

[129] A. Takhom, S. Usanavasin, T. Supnithi and P. Boonkwan, A collaborative framework supporting ontology development based on agile and scrum model, *IEICE Trans. Inf. & Syst.* **E 103.D** (2020), 2568–2577. doi:10.1587/transinf.2020EDP7041.

[130] H. Tan, A. Adlemo, V. Tarasov and M.E. Johansson, Evaluation of an application ontology, in: *The Joint Ontology Workshops*, 2017.

[131] L. Tankeleviciene and R. Damasevicius, Characteristics of domain ontologies for web based learning and their application for quality evaluation, *Informatics in Education* **8** (2009), 131–152. doi:10.15388/infedu.2009.09.

[132] S. Tartir, I.B. Arpinar, M. Moore, A.P. Sheth and B. Aleman-Meza, *OntoQA: Metric-Based Ontology Quality Analysis, IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources*, 2005, https://corescholar.libraries.wright.edu/knoesis/660.

[133] S. Tartir, I.B. Arpinar and A.P. Sheth, Ontological evaluation and validation, in: *Theory and Applications of Ontology: Computer Applications*, R. Poli, M. Healy and A. Kameas, eds, Springer, Dordrecht, 2010, pp. 115–130. doi:10.1007/978-90-481-8847-5_5.

[134] W. Terkaj, S. Borgo and E. Sanfilippo, Ontology for industrial engineering: A DOLCE compliant approach, in: *CEUR Workshop Proceedings*, Vol. 3240, 2022.

[135] The OWL Working Group OWL Web Ontology Language Guide, 2004, https://www.w3.org/TR/owl-guide/ (accessed December 29, 2021).

[136] The RDF Working Group, Resource Description Framework (RDF): Concepts and Abstract Syntax, 2004, https://www.w3.org/TR/rdf-concepts/ (accessed December 29, 2021).

[137] D. Tsarkov, Improved algorithms for module extraction and atomic decomposition, in: *The International Workshop on Description Logics (DL'12)*, Rome, Italy, 2012, pp. 7–10.

[138] M. Uschold and M. Gruninger, Ontologies: Principles, methods and applications, *The Knowledge Engineering Review* **11** (1996), 93–136. doi:10.1017/S0269888900007797.

[139] M. Uschold and M. King, Towards a methodology for building ontologies, in: *IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing*, Montreal, 1995.

[140] A. Verma, An abstract framework for ontology evaluation, in: *International Conference on Data Science and Engineering (ICDSE)*, 2016, pp. 1–6. doi:10.1109/ICDSE.2016.7823945.

[141] A. Vizedom, OntologySummit2013_From-Use-Context-to-Evaluation–AmandaVizedom_20130228.pdf, http://ontolog-archives.cim3.net/file/work/OntologySummit2013/2013-02-28_OntologySummit2013_OntologyEvaluation-ExtrinsicAspects-2/OntologySummit2013_From-Use-Context-to-Evaluation--AmandaVizedom_20130228.pdf (accessed June 26, 2021).

[142] D. Vrandečić, Ontology evaluation, in: *Handbook on Ontologies*, Springer, 2009, pp. 293–313. doi:10.1007/978-3-540-92673-3_13.

[143] D. Vrandečić and A. Gangemi, Unit tests for ontologies, in: *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, R. Meersman, Z. Tari and P. Herrero, eds, Springer, Berlin, Heidelberg, 2006, pp. 1012–1020. doi:10.1007/11915072_2.

[144] W3C RDF Validation Service, 2006, https://www.w3.org/RDF/Validator/ (accessed December 29, 2021).

[145] A.I. Walisadeera, A. Ginige and G.N. Wikramanayake, User centered ontology for Sri Lankan farmers, *Ecological Informatics* **26** (2015), 140–150. doi:10.1016/j.ecoinf.2014.07.008.

[146] R.Y. Wang and D.M. Strong, Beyond accuracy: What data quality means to data consumers, *Journal of Management Information Systems* **12** (1996), 5–33. doi:10.1080/07421222.1996.11518099.

[147] R.S.I. Wilson, J.S. Goonetillake, W.A. Indika and A. Ginige, Analysis of ontology quality dimensions, criteria and metrics, in: *Computational Science and Its Applications – ICCSA 2021*, O. Gervasi, B. Murgante, S. Misra, C. Garau, I. Blečić, D. Taniar, B.O. Apduhan, A.M.A.C. Rocha, E. Tarantino and C.M. Torre, eds, Springer International Publishing, Cham, 2021, pp. 320–337. doi:10.1007/978-3-030-86970-0_23.

[148] S.I. Wilson, J.S. Goonetillake, A. Ginige and A.I. Walisadeera, Towards a usable ontology: The identification of quality characteristics for an ontology-driven decision support system, *IEEE Access* **10** (2022), 12889–12912. doi:10.1109/ACCESS.2022.3146331.

[149] J. Yu, J.A. Thom and A. Tam, Ontology evaluation using Wikipedia categories for browsing, in: *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management (CIKM '07)*, Association for Computing Machinery, New York, NY, USA, 2007, pp. 223–232. doi:10.1145/1321440.1321474.

[150] J. Yu, J.A. Thom and A. Tam, Requirements-oriented methodology for evaluating ontologies, *Information Systems* **34**(8) (2009), 766–791. doi:10.1016/j.is.2009.04.002.

[151] A. Zaveri, D. Kontokostas, M.A. Sherif, L. Bühmann, M. Morsey, S. Auer and J. Lehmann, User-driven quality evaluation of DBpedia, in: *Proceedings of the 9th International Conference on Semantic Systems*, Association for Computing Machinery, New York, NY, USA, 2013, pp. 97–104. doi:10.1145/2506182.2506195.

[152] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann and S. Auer, Quality assessment for linked data: A survey, in: *Semantic Web*, IOS Press, 2016, pp. 63–93. doi:10.3233/SW-150175.

[153] H. Zhu, D. Liu, I. Bayley, A. Aldea, Y. Yang and Y. Chen, Quality model and metrics of ontology for semantic descriptions of web services, *Tsinghua Science and Technology* **22** (2017), 254–272. doi:10.23919/TST.2017.7914198.

[154] X. Zhu, J.W. Fan, D.M. Baorto, C. Weng and J.J. Cimino, A review of auditing methods applied to the content of controlled biomedical terminologies, *J. Biomed. Inform.* **42**(3) (2009), 413–425. doi:10.1016/j.jbi.2009.03.003.